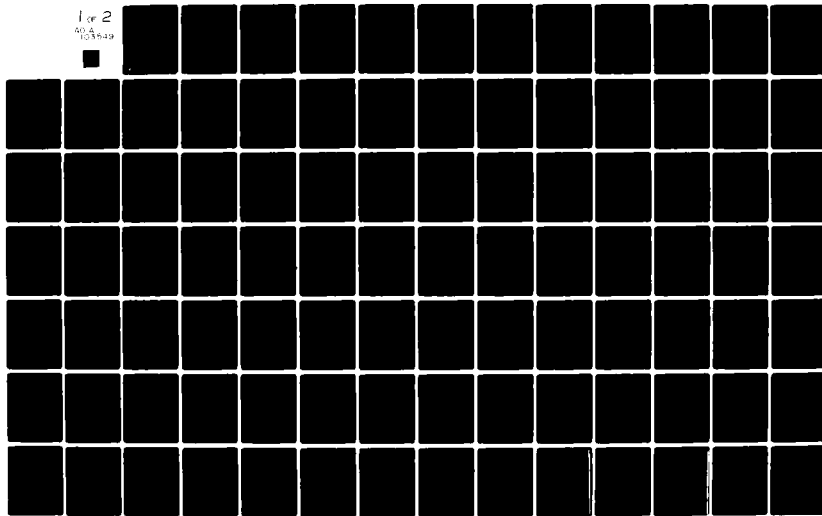


AD-A103 549

PENNSYLVANIA STATE UNIV UNIVERSITY PARK APPLIED RESE--ETC F/G 20/4
A COMPUTER PROGRAM FOR CALCULATING POTENTIAL FLOW SOLUTIONS FOR--ETC(U)
JUN 81 A M YOCUM N00024-79-C-6843
ARL/PSU/TH-81-130 NL

UNCLASSIFIED

1 of 2
00 A
103549



AD A103549

LEVEL II

⑥ B

A COMPUTER PROGRAM FOR CALCULATING POTENTIAL FLOW
SOLUTIONS FOR FLOW THROUGH LINEAR AND STATIONARY
CIRCULAR CASCADES

A. M. Yocum

Technical Memorandum

File No. TM 81-130

26 June 1981

Contract No. N00024-79-C-6043

(Copy No. 15/16)

The Pennsylvania State University
APPLIED RESEARCH LABORATORY
Post Office Box 30
State College, PA 16801

Approved for Public Release
Distribution Unlimited

NAVY DEPARTMENT

NAVAL SEA SYSTEMS COMMAND

DTIC
ELECTE
SEP 1 1981
A

DTIC FILE COPY

81 9 01 122

= 91004

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TM 81-130	2. GOVT ACCESSION NO. ✓ AD-A103549	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A COMPUTER PROGRAM FOR CALCULATING POTENTIAL FLOW SOLUTIONS FOR FLOW THROUGH LINEAR AND STATIONARY CIRCULAR CASCADES	5. TYPE OF REPORT & PERIOD COVERED Technical Memorandum	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) A. M. Yocum	8. CONTRACT OR GRANT NUMBER(s) N00024-79-C-6043 ✓	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Applied Research Laboratory Post Office Box 30 State College, PA 16801	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Department of the Navy Washington, DC 20362	12. REPORT DATE 26 June 1981	
	13. NUMBER OF PAGES 132	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release. Distribution unlimited Per NAVSEA - August 11, 1981.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) flow, circular, cascades, computer program		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) ✓ The Douglas Neumann Cascade Program has been rewritten so that it can analyze circular cascades with inward or outward radial flow directions, in addition to linear cascades for which it was originally developed. A solution for circular cascades is obtained by employing a conformal transformation which maps the circular cascade into a linear cascade. The capability available in the original program of analyzing multiple cascades has been preserved and can also be applied to circular cascades. In addition, the use of auxiliary storage		

20. has been modified and a new procedure for solving the system of equations for large problems has been developed which was found to significantly reduce the computation time. Included in this report is an explanation of the theory and a description of the program and its verification. Appendix A is a users guide for running the program.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

Subject: A Computer Program for Calculating Potential Flow Solutions
for Flow Through Linear and Stationary Circular Cascades

References: See page 70.

Abstract: The Douglas Neumann Cascade Program has been rewritten so that it can analyze circular cascades with inward or outward radial flow directions, in addition to linear cascades for which it was originally developed. A solution for circular cascades is obtained by employing a conformal transformation which maps the circular cascade into a linear cascade. The capability available in the original program of analyzing multiple cascades has been preserved and can also be applied to circular cascades. In addition, the use of auxiliary storage has been modified and a new procedure for solving the system of equations for large problems has been developed which was found to significantly reduce the computation time. Included in this report is an explanation of the theory and a description of the program and its verification. Appendix A is a users guide for running the program.

Acknowledgement: This project was sponsored by the Hydro-Turbine Division of the Allis-Chalmers Corporation.

Table of Contents

	<u>Page</u>
Abstract	1
Acknowledgement	1
List of Figures	3
List of Tables	4
Nomenclature	6
I. INTRODUCTION	9
II. THEORY	12
2.1 Basic Equations	12
2.2 Basic Solutions and Combination Equations	16
2.3 Transformation of Circular Cascades	23
III. PROGRAM DESCRIPTION	26
3.1 Definition of Key Input and Output Variables	26
3.2 Cascade Modeling	34
3.3 Structure of the Program	41
IV. PROGRAM VERIFICATION	47
4.1 Methods of Verification	47
4.2 Exact Analytical Cascade Solutions	49
4.3 Comparisons of the Program Results With Exact Analytical Solutions	60
4.4 Verification of Results for Multiple Cascades of Lifting and Nonlifting Bodies	68
REFERENCES	70
APPENDIX A: A User's Guide for Running the Program	71
A.1 Explanation of Program Input	71
A.2 Sample Input	82
A.3 Sample Output	87
APPENDIX B	102
B.1 Data Storage and Equation Solution Procedures	102
B.2 Changing Array Dimensions and Solution Procedures	115
APPENDIX C: Coordinate Data Input Order and Reordering	125

List of Figures

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
1	A Typical Straight Line Element of the Body Surface	15
2	Segments of Two Infinite Cascades With Velocity Triangles Illustrating the Inlet, Average and Exit Velocities and Flow Angles	17
3	Schematic of a Circular Cascade Which is Transformed Into the Linear Cascade	24
4	Illustration of How a Cascade of Cylinders With Unequal Spacing Can be Modeled as Two Cascades of Equal Spacing SP, Offset by the Distance ADDY	36
5	Wicket Gates and Stay Vanes of a Hydraulic Turbine	38
6	Illustration of One Possible Way to Model the Wicket Gates and Stay Vanes in the Region of Stay Vane D of Figure 5	40
7	Illustration of the Transformation of a Cylinder in the S-Plane to the Two-Dimensional Linear Cascade in the Z-Plane and the Addition Transformation of the Linear Cascade to the Circular Cascade in the T-Plane. The Radial Flow is Outward in the T-Plane	50
8	Illustration of the Transformations Which Yield a Circular Cascade in the T-Plane With the Radial Flow Inward	54
9	A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Analytically Derived Airfoil Shape in a Linear Cascade. The Cascade Geometry is Illustrated in the Z-Plane of Figure 7	61
10	A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Circular Cascade Illustrated in the T-Plane of Figure 7	65
11	A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Circular Cascade Illustrated in the T-Plane of Figure 8	67
12	A Comparison of Pressure Distributions for Cascades of Lifting and Non-Lifting Bodies Obtained From the Original and the Current Version of the Douglas Neumann Cascade Program	69
A.1	Schematic of Two Linear Cascades Illustrating the Geometric Input Parameters for the Program	73

List of Figures (Cont'd)

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
A.2	Schematic of Two Circular Cascades Illustrating the Geometric Input Parameters for the Program. The Cascade Geometric Input are Independent of the Radial Flow Direction	74
B.1	Illustration of How a Block of Equations Represented by a Portion of a 16 x 16 Matrix is Partially Reduced in the Core. The X's Represent Non-zero Numbers	107
B.2	Schematic of a 16 x 16 Matrix After it Has Been Partially Reduced by the First Four Rows of the Matrix	108
B.3	Schematic of the Example 16 x 16 Matrix After the Second Step of the Reduction. The Brackets Show the Block of the Matrix Which Was in the Core for the Second Step	110
B.4	A Listing of the Main Program	116
B.5	Parts of Subroutines PART4 and MIS1 Showing the Array Dimensions	120
B.6	Beginning of SUBROUTINE PART3 Showing the COMMON Statements and Important Parameters	121
B.7	Beginning of SUBROUTINE PART5 Showing the Array Dimensions . .	123
C.1	An Airfoil in its Local Coordinate System x'-y', with the Order for the Data Input Indicated by the Arrows	127
C.2	A Linear Cascade Showing the Orientation of Airfoil Coordinate System With Respect to the Cascade Coordinate System	128
C.3	a. Schematic of a Circular Cascade Showing the Orientation of the Airfoil Coordinate System in the Global System. b. A Schematic Showing the Cascade Orientation and Coordinate Input Direction After the Transformation. c. Final Cascade Configuration and Order of Coordinate Data	129
C.4	a. Schematic of a Circular Cascade with the Radial Flow Direction Outward. The x'-y' Orientation is the Same as in Figure C3a. b. Final Configuration of the Cascade After the Transformation	132

List of Tables

<u>Table No.</u>	<u>Title</u>	<u>Page</u>
1	A Comparison of Cascade Parameters Obtained From the Program for the Various Input Parameters	63
2	A Comparison of Velocity Components at Off-Body Points Obtained From the Cascade Program and the Exact Analytical Solution	64
B.1	Steps in Reducing a 200 x 200 Matrix in a 50 x 50 Array . . .	112
B.2	A Comparison of the Computation Times Required by the Three Solution Procedures for a Sample Problem	112
B.3	A Comparison of the Three Solution Procedures	114

Nomenclature

- a - radius of the cylinder which is transformed into a cascade
- [a] - influence coefficient matrix
- A_{jk} - the normal component of velocity induced at the midpoint of element j by the source distribution on element k of constant unit strength
- A'_{jk} - the normal component of velocity induced at the midpoint of element j by a distribution of constant unit strength vortices on element k
- $A(q,s) - \vec{V}(q,s) \cdot \vec{n}(s)$
- c - chord of the airfoils
- C_{F_x} - force coefficient in nondimensionalized form defined by Eq. 47
- C_{F_y} - force coefficient in nondimensionalized form defined by Eq. 48
- C_L - the total lift coefficient defined by Eq. 45
- C_M - moment coefficient defined by Eq. 49
- C_p - pressure coefficient defined by Eq. 43
- d - distance to the point where the cylinder intersects the x axis in the S plane for the exact cascade flow solution
- E - source and sink strength in the exact cascade flow solution
- F_L - sum of the lift forces for one airfoil of each cascade
- F_r - force on the airfoil in the r direction
- F_θ - force on the airfoil in the θ direction
- F_x - force on the airfoil in the global x direction
- F_y - force on the airfoil in the global y direction
- G - vortex strength at A' and A'' in the exact cascade flow solution
- H - vortex strength at B' and B'' in the exact cascade flow solution
- K_1 - $-E + iG$
- \bar{K}_1 - complex conjugate of K_1
- K_2 - $E - iG$
- \bar{K}_2 - complex conjugate of K_2

Nomenclature (Cont'd)

M	- moment about a point specified in the input
N	- number of equations or elements
NC	- number of cascades with circulation
n_1	- indices of the first and last elements of the
n_2	- airfoils in the m cascade
$n(s)$	- unit vector normal to the surface at point S
$[0]$	- onset flow matrix
P_{s_l}	- local static pressure
$P_{s_{ref}}$	- reference static pressure
P_T	- total pressure, constant throughout the potential flow field
Q	- volume flow rate
r_{ref}	- a reference radius specified by input to the cascade program
S	- complex coordinate in the S -plane
S_m	- the circumference of an airfoil in cascade m
SP	- dimensionless cascade spacing
U	- magnitude of the average velocity vector
V	- velocity vector
V_{ref}	- U for linear cascades, $V_{r_{ref}}$ for circular cascades
V_∞	- the onset flow
$V(q,s)$	- velocity at point s due to an infinite array of unit sources at point q on the bodies
V_{dn}	- downwash velocity
V_l	- local velocity
V_{up}	- upwash velocity
ΔV_{0_m}	- cascade m 's violation of the Kutta condition for 0 degree onset flow

Nomenclature (Cont'd)

- ΔV_{90_m} - cascade m's violation of the Kutta condition for the 90 degrees onset flow
- $\Delta V_{m,n}$ - cascade m's violation of the Kutta condition for the onset flow created by the circulation on cascade n
- W - the width of the radial channel
- $W(S)$ - complex velocity in the S-plane
- $W(Z)$ - complex velocity in the Z-plane
- $W(T)$ - complex velocity in the T-plane
- z_j - the complex coordinate of the midpoint of element j
- $z(s)$ - complex coordinate at point $s = x + iy$
- $\zeta(q)$ - complex coordinate at point $q = \zeta + in$
- $\sigma(q)$ - source strength at point q
- α_I - inlet flow angle
- $\Delta\alpha$ - the angle the flow is turned through the cascade
- α - the mean flow angle
- α_E - exit flow angle
- $[\sigma]$ - source distribution matrix
- ψ - stream function
- Γ_m - the circulation on an airfoil in cascade m

I. INTRODUCTION

The original Douglas Neumann Cascade Program, documented in a report by Joseph P. Geising [1]*, solves an approximate potential solution for the flow through two-dimensional linear cascades. This approximate solution approaches the exact solution as the number of points describing the body surface approaches infinity. The original program can handle one or more cascades, the bodies can be of any arbitrary geometry, and the bodies can be lifting or nonlifting.

The Applied Research Laboratory of The Pennsylvania State University modified the Douglas Neumann program so that flow inward through a single radial flow or circular cascade could be analyzed. This modified program was developed under sponsorship by the U.S. Bureau of Reclamation to enable the flow through wicket gates of a hydraulic turbine to be analyzed [2]. A conformal transformation was incorporated into the program which transformed the circular cascade into a linear cascade. The transformed cascade could then be analyzed in the conventional manner and the solution then transformed back to the real plane. In Reference [2], average downstream flow angles calculated with the modified program are compared to experimental values obtained in an air test facility. The agreement between the experimental and calculated flow angles is found to be quite good.

When the work of [2] was performed, much of the flexibility of the original Douglas Neumann Cascade program was lost, and also, both linear and circular cascades could not be analyzed with a single program. This report documents a rewritten version of the Douglas Neumann program which can handle both linear and circular cascades, multiple cascades of both types can be analyzed, and the input parameters which define the flow

*Numbers in brackets refer to references at the end of the report.

conditions through the cascades can be of several different types as in the original program. In addition, the new program can analyze circular cascades with the radial flow direction inward or outward. Thus, for example, the flow through stay vanes and wicket gates of a pump-turbine can be analyzed in both the pump and turbine mode.

With the many different types of problems to be handled by a single program, it was thought that the program would be more orderly if it was rewritten following the original technique, rather than inserting a lot of new sections into the old program. The program is now structured so that data are read in, manipulated, calculations are performed etc. in progressive steps through the program. There are numerous sections of the program which will not apply to a particular problem, thus the program skips these sections and continues on with the next section which applies. The program does not double back over itself, except in small iteration loops, and, thus, the logic should be easy to follow. The functions of the various sections of the new program are also documented with comment statements throughout the program.

In the Douglas Neumann Cascade program, the flow can be analyzed at points off the body in addition to obtaining the flow on the body surface and the overall cascade performance. In many circumstances it may be desirable to analyze a large number of off-body points which are systematically spaced through the cascades. The rewritten version of the program will generate off-body point coordinates given an initial set of coordinates and the desired spacing. In the new version, the off-body point coordinates are also stored in arrays separate from the body coordinates so that the number of off-body points need not be considered when selecting the number of points to represent the airfoils. Two

large arrays have been specified for storing off-body point coordinates and the separate storage makes changing the dimensions of the arrays an easy task, if it becomes necessary.

The use of auxiliary and core storage has also been greatly modified in the new version of the program. The original Douglas Neumann program used tapes to store data and transfer the data from subroutine to subroutine. This was probably done so that only one subroutine needed to be in the core at one time, and thus the program could be run on smaller computers. With today's larger computers, COMMON storage can be used for transferring data between subroutines, and the repetitious reading of data from auxiliary storage eliminated. Auxiliary storage is currently only used to store the original body coordinates and the very large matrices representing the system of equations which are solved simultaneously.

For very large problems, the matrix representing the system of equations to be solved is too large to fit in the core of most computers. The original Douglas Neumann program utilized an iterative technique and auxiliary storage to solve the equations for large problems. The iterative approach required reading the equations from a file for each iteration. Since reading of disc files is time consuming, a new procedure for solving the equations was developed during the current rewriting of the program. Both the new procedure and the iterative procedure are available with the rewritten version of the program. The new procedure was shown in a sample run to significantly reduce the computation time.

Thus far, this introduction has given a general description of the background and development of the cascade program. The major changes made in the rewritten version of the Douglas Neumann cascade program have also been briefly described. In the remainder of this report, the theory used

in the program, a description of the program and the input and output parameters, and several check cases will be presented. A users guide giving detailed instructions for running the program and a sample run can be found in Appendix A. Appendix B discusses the data storage and solution procedures, and Appendix C describes the coordinate data input order and the necessary reordering performed in the program.

II. THEORY

2.1 Basic Equations

The basic concept of the Douglas Neumann cascade program is to apply a distribution of sources on the surface of the bodies such that the combination of the source distribution and the onset flow satisfy the boundary condition on the blade surface. For the usual condition of no flow normal to the surface, the following equation should be satisfied everywhere on the blade surface:

$$-\vec{V}_{\infty} \cdot \vec{n}(s) = \int_{\text{body}} \sigma(q) A(q,s) dq \quad (1)$$

where

\vec{V}_{∞} = the onset flow

$\vec{n}(s)$ = unit vector normal to the surface at point s

$\sigma(q)$ = source strength at point q

$A(q,s) = \vec{V}(q,s) \cdot \vec{n}(s)$

$\vec{V}(q,s)$ = velocity at point s due to an infinite array of unit sources at point q on the bodies.

An expression for $\vec{V}(q,s)$ can be obtained by starting with the equation for the velocity at a point due to a single source and writing $\vec{V}(q,s)$ as an infinite sum of the velocities induced by an infinite array of

sources. The infinite sum can then be eliminated by manipulating the equation and introducing the hyperbolic functions. The following equation is obtained for $\vec{V}(q,s)$:

$$\vec{V}(q,s)_{\text{source}} = V_x - iV_y = \frac{1}{2SP} \coth \left\{ \frac{\pi}{SP} [z(s) - \zeta(q)] \right\} \quad (2)$$

where

$z(s)$ = complex coordinate at point $s = x + iy$

$\zeta(q)$ = complex coordinate at point $q = \xi + i\eta$

The expression for the velocity induced by an infinite array of vortices is the same as Equation (2), but with the velocity vector rotated 90° .

$$\vec{V}(q,s)_{\text{vortex}} = i \frac{1}{2SP} \coth \left\{ \frac{\pi}{SP} [z(s) - \zeta(q)] \right\} \quad (3)$$

The technique employed in the Douglas Neumann program to solve Equation (1) is to; (1) break the surface of the bodies into small segments or elements, (2) approximate the source distribution as a constant on each element, and (3) satisfy the boundary condition only at the element midpoint. With these approximations, Equation (1) can be written as:

$$\vec{V}_{\infty j} \cdot \vec{n}_j = \sum_{k=1}^N \sigma_k \int_{\text{element } k} A_j(q) d(q) \quad (4)$$

As seen in Equation (4), the independent variable s has been eliminated, because Equation (4) is evaluated only at the midpoints of the elements. Thus, $A(q,s)$ becomes $A_j(q)$ and can be written as:

$$A_j(q) = \vec{V}_j(q) \cdot \vec{n}_j = \left[\frac{1}{2SP} \coth \left\{ \frac{\pi}{SP} [z_j - \zeta(q)] \right\} \right] \cdot \vec{n}_j \quad (5)$$

where

z_j = the complex coordinate of the midpoint of element j .

The integral in Equation (4) can be evaluated by expressing the surface coordinate q in terms of the global coordinates of the body in the cascade. With the aid of Figure 1 we find that:

$$\zeta(q) = (\xi_o + i n_o) + q(\cos \beta_k + i \sin \beta_k) = C_{o_k} + qe^{i\beta_k} \quad (6)$$

Equation (6) can then be differentiated to yield an expression for dq , which is needed to perform the integration.

$$dq = d\zeta e^{-i\beta_k} \quad (7)$$

Substituting Equations (5) and (7) into the integral and expressing the integration limits as the end points of element k as defined in Figure 1 yields:

$$\int_{\text{element } k} A_j(q) dq = \vec{n}_j \cdot \int_{C_{1_k}}^{C_{2_k}} \frac{1}{2SP} \coth \left[\frac{\pi}{SP} (z_j - \zeta) \right] d\zeta e^{-i\beta_k} \quad (8)$$

Carrying out the integration in Equation (8), provides an expression for A_{jk} , the normal velocity induced at the midpoint of element j due to the source distribution on element k of constant unit strength.

$$A_{jk} \equiv \int_{\text{element } k} A_j(q) dq = \frac{e^{-i\beta_k}}{2\pi} \ln \left[\frac{\sinh \frac{\pi}{SP} (z_j - C_{1_k})}{\sinh \frac{\pi}{SP} (z_j - C_{2_k})} \right] \cdot \vec{n}_j \quad (9)$$

Substituting A_{jk} as defined in Equation (9) into Equation (4) yields the basic equation used in the Douglas Neumann program.

$$\sum_{k=1}^N A_{jk} \sigma_k = -\vec{V}_{\infty j} \cdot \vec{n}_j \quad (10)$$

Equation (10) is the approximate form of Equation (1) and it must be satisfied for each element of the cascades. Equation (10) can be written for each element yielding the following matrix equation:

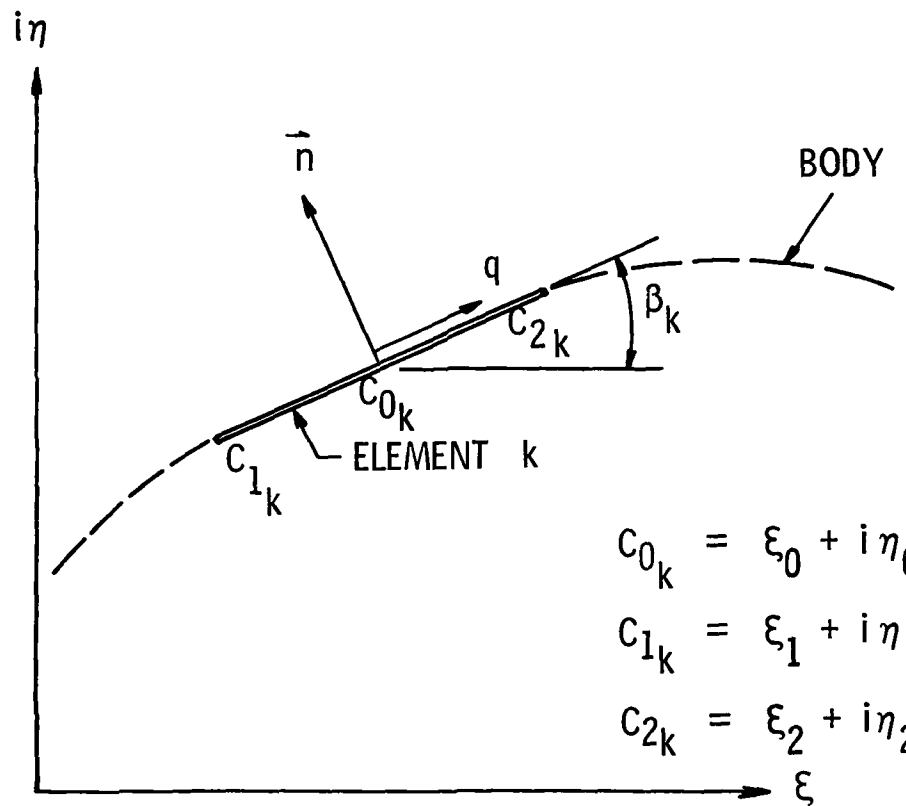


Figure 1. A Typical Straight Line Element of the Body Surface

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ A_{N1} & A_{N2} & & A_{NN} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \cdot \\ \cdot \\ \cdot \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \vec{-V}_{\infty 1} & \cdot & \vec{n}_1 \\ \vec{-V}_{\infty 2} & \cdot & \vec{n}_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \vec{-V}_{\infty N} & \cdot & \vec{n}_N \end{bmatrix} \quad (11)$$

The set of equations represented by Equation (11) are solved simultaneously in the Douglas Neumann program for the unknown source strength of each element. As seen by examining Equation (9), the terms of the A matrix are only functions of element coordinates and, thus, can be calculated from the cascade geometries which are input to the program. The onset flows, appearing in the matrix on the right hand side of Equation (11), will be discussed in the following section.

2.2 Basic Solutions and Combination Equations

Equation (11) can be solved for any given onset flow V_{∞} . However, for airfoils in a cascade, an addition boundary condition, must also be satisfied. Circulation, which arises around the airfoils, enables the Kutta condition to be met. For cascades, this circulation induces an upwash and downwash velocity far upstream and downstream, respectively, relative to the mean cascade velocity. These induced velocities are illustrated in Figure 2. Both the circulation and the mean velocity must be accounted for in Equation (11). However, for the typical cascade problem, the magnitude of the circulation is usually unknown and the mean flow angle may not be given. The operating parameter which may be given for a cascade analysis will be either:

1. α_I , the inlet flow angle
2. $\Delta\alpha$, the angle the flow is turned through the cascade = $\alpha_I - \alpha_E$

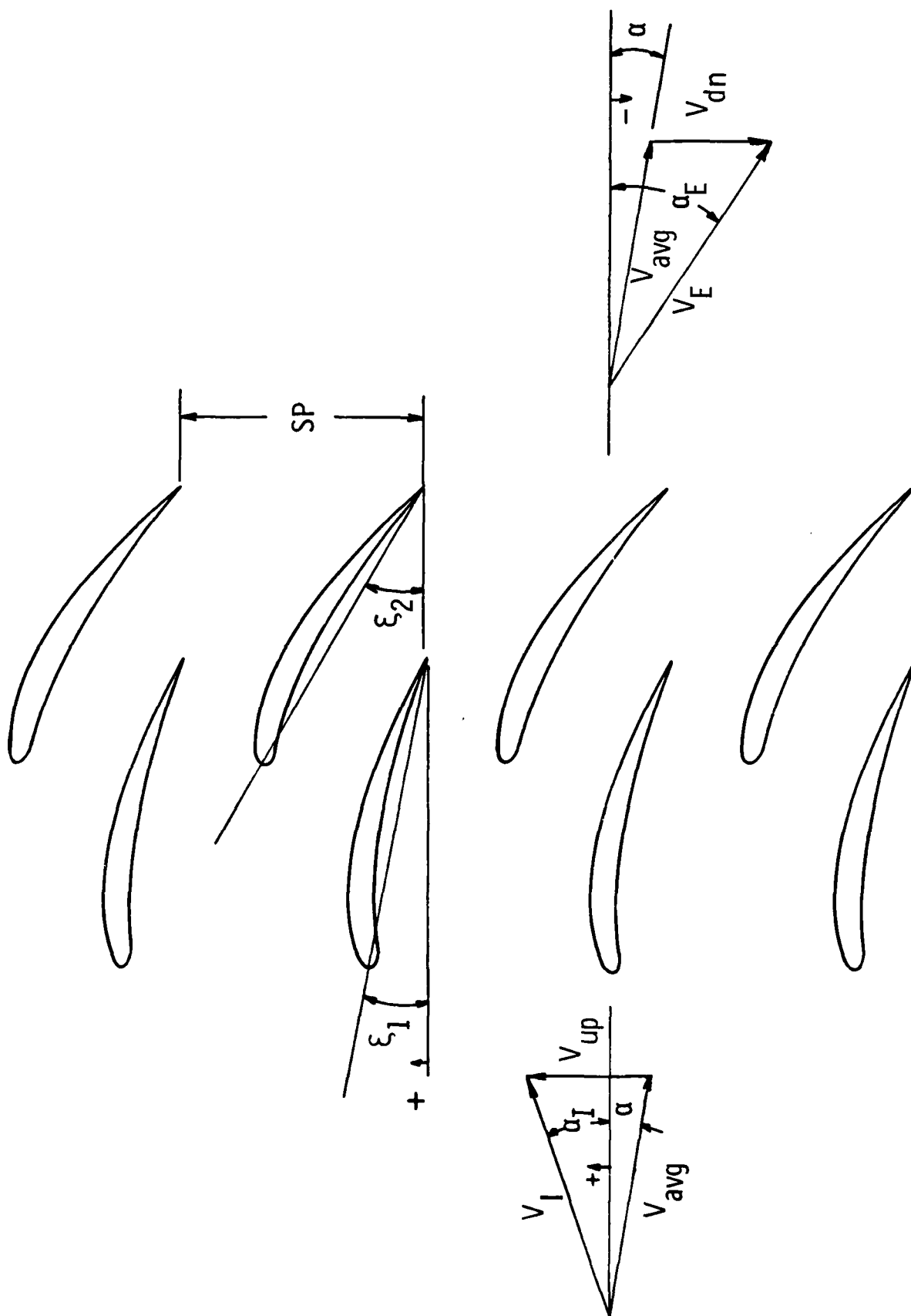


Figure 2. Segments of Two Infinite Cascades With Velocity Triangles Illustrating the Inlet, Average and Exit Velocities and Flow Angles

3. α , the mean flow angle
 - or 4. C_L , the total lift coefficient of all the cascades.
- The angles α_I , α and α_E are shown in Figure 2.

In the Douglas Neumann program, several basic solutions are calculated, each of which satisfy the condition of zero velocity normal to the body surfaces. Since the governing equation is linear, any number of solutions can be scaled and added without violating the governing equation or the surface boundary condition. The technique employed by the Douglas Neumann program is to combine the basic solutions such that the cascade operating parameter and Kutta condition are satisfied.

The basic solutions which are used are calculated for each of the following onset flows:

1. A uniform flow normal to the cascade(s), i.e., $\alpha = 0$
2. A uniform flow parallel to the cascade(s), i.e., $\alpha = 90^\circ$
3. A nonuniform flow created by the circulation on the airfoils of cascade 1.
4. A nonuniform flow, like flow 3, for each additional cascade with circulation.

The terms $\vec{v}_{\infty j} \cdot \vec{n}_j$ in Equation (11) for onset flows 1 and 2 are simply $\sin(\beta_j)$ and $-\cos(\beta_j)$, where β is the angle of the element as previously illustrated in Figure 1. For the nonuniform onset flows, the effects of all the elements of an airfoil in one cascade on a particular element must be summed to obtain the velocity induced by the circulation. Since the only difference in the velocity induced by a vortex or a source distribution is that the velocity vector is rotated 90 degrees, the calculation performed in the program to evaluate Equation (9) can also be used to calculate the onset flows created by the circulation of the different

cascades. The normal component of the onset flow for element j, due to circulation on the airfoils in cascade m can be written as:

$$\vec{V}_{\infty j} \cdot \vec{n}_j = \frac{\Gamma_m}{S_m} \sum_{k=n_1}^{n_2} A'_{jk} \quad (12)$$

where Γ_m = the circulation on cascade m

S_m = the circumference of an airfoil in cascade m

A'_{jk} = the normal component of the velocity induced at the midpoint of element j by a distribution of constant unit strength vortices on element k. (Since the dot product in Equation (9) is performed by complex number multiplication, A'_{jk} is obtained from the real part of the resulting number while A_{jk} is the imaginary part.)

n_1 and n_2 = the indices of the first and last elements of the airfoils in the m cascade.

The value of Γ_m in Equation (12) is unknown at the time the basic solutions are calculated. Thus, the basic solutions for the onset flows due to circulation are obtained with $\Gamma_m = 1$. The real magnitude of Γ_m is determined when the basic solutions are combined to satisfy the Kutta condition.

Under most circumstances, the Kutta condition is violated by each of the basic solutions. An approximate measure of the amount each solution violates the Kutta condition can be obtained by computing the difference in the tangential velocity above and below the trailing edge (i.e. at the midpoint of the first and last element of each cascade). The basic solutions must be combined such that the cascade operating parameter is met and the combined violations of the Kutta condition are zero. The

following equations express these conditions and provide a means for evaluating the unknown values of Γ_m .

$$\begin{aligned} \Delta V_{0_1} U \cos \alpha + \Delta V_{90_1} U \sin \alpha + \Delta V_{11} \Gamma_1 + \Delta V_{12} \Gamma_2 + \dots &= 0 \\ \Delta V_{0_2} U \cos \alpha + \Delta V_{90_2} U \sin \alpha + \Delta V_{21} \Gamma_1 + \Delta V_{22} \Gamma_2 + \dots &= 0 \\ \cdot &\cdot \\ \cdot &\cdot \\ \cdot &\cdot \\ \Delta V_{0_{NC}} U \cos \alpha + \Delta V_{90_{NC}} U \sin \alpha + \Delta V_{NC,1} \Gamma_1 + \Delta V_{NC,2} \Gamma_2 + \dots &= 0 \end{aligned} \quad (13)$$

where:

ΔV_{0_m} = cascade m's violation of the Kutta condition for the 0 degree onset flow.

ΔV_{90_m} = cascade m's violation of the Kutta condition for the 90 degree onset flow.

$\Delta V_{m,n}$ = cascade m's violation of the Kutta condition for the onset flow created by the circulation on cascade n.

NC = number of cascades with circulation.

α = cascade average flow angle = $\tan^{-1} \left(\frac{\tan \alpha_I + \tan \alpha_E}{2} \right)$.

If α is the given cascade operating parameter, Equation (14) can be solved for the unknown values of Γ_m . For this case, Equation (13) is rewritten in matrix form, Equation (14), which can then be solved by matrix reduction. For the usual case of one cascade, the solution is trivial.

$$\begin{bmatrix} \Delta V_{11} & \Delta V_{12} & \dots & \Delta V_{1,NC} \\ \Delta V_{21} & \Delta V_{22} & \dots & \Delta V_{2,NC} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \Delta V_{NC,1} & \Delta V_{NC,2} & \dots & \Delta V_{NC,NC} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \cdot \\ \cdot \\ \cdot \\ \Gamma_{NC} \end{bmatrix} = -U \begin{bmatrix} \Delta V_{01} \cos \alpha + \Delta V_{901} \sin \alpha \\ \Delta V_{02} \cos \alpha + \Delta V_{902} \sin \alpha \\ \cdot \\ \cdot \\ \cdot \\ \Delta V_{0NC} \cos \alpha + \Delta V_{90NC} \sin \alpha \end{bmatrix} \quad (14)$$

For the cases when α is not the given operating parameter, Equation (14) cannot be used directly to solve for the Γ_m 's; because, explicit relations do not exist between α and α_I , C_L , or $\Delta\alpha$. When α_I is the given input parameter, a matrix equation which contains the additional unknown α is solved. For the other cases when C_L or $\Delta\alpha$ are given, the program must iterate to obtain the additional unknown α .

To obtain the matrix equation which contains α as an unknown and α_I as a known parameter, an additional equation is required. By examining the inlet velocity triangle in Figure 2, the following equation can be obtained:

$$U \cos \alpha \tan \alpha_I - U \sin \alpha = V_{up} \quad (15)$$

The upwash velocity, V_{up} , in Equation (15) is unknown. However, since V_{up} is the induced velocity due to the circulation on the airfoils, it is known that $\sum \Gamma_m = 2SP V_{up}$. The upwash velocity can therefore be eliminated from Equation (15) and after rearranging it yields the following:

$$\tan \alpha + [\Gamma_1/U \cos \alpha + \Gamma_2/U \cos \alpha \dots \Gamma_{NC}/U \cos \alpha]/2SP = \tan \alpha_I \quad (16)$$

Modifying Equation (14) algebraically and inserting Equation (16) provides the necessary matrix equation which can be solved for α and the Γ_m 's for a given α_I .

$$\begin{bmatrix} U\Delta V90_1 & \Delta V_{11} & \Delta V_{12} & \dots & \Delta V_{1,NC} \\ U\Delta V90_2 & \Delta V_{21} & \Delta V_{22} & \dots & \Delta V_{2,NC} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ U\Delta V90_{NC} & \Delta V_{NC,1} & \Delta V_{NC,2} & & \Delta V_{NC,NC} \\ 1 & \frac{1}{2SP} & \frac{1}{2SP} & & \frac{1}{2SP} \end{bmatrix} \begin{bmatrix} \tan \alpha \\ \Gamma_1/U \cos \alpha \\ \\ \Gamma_{NC-1}/U \cos \alpha \\ \Gamma_{NC}/U \cos \alpha \end{bmatrix} = \begin{bmatrix} -U\Delta V0_1 \\ -U\Delta V0_2 \\ \cdot \\ \cdot \\ \cdot \\ -U\Delta V0_{NC} \\ \tan \alpha_I \end{bmatrix} \quad (17)$$

If $\Delta\alpha$ or C_L are given as the input parameters, the program will iterate using an assumed value of α in Equation (14), until the α is found which yields the given $\Delta\alpha$ or C_L . Therefore, equations which relate C_L and $\Delta\alpha$ to the Γ_m 's found for a particular α are required. These equations are:

$$C_L = \frac{2}{Uc} \sum_{m=1}^{NC} \Gamma_m \quad (18)$$

and

$$\tan \Delta\alpha = \frac{\frac{\cos \alpha}{SP} \sum_{m=1}^{NC} \Gamma_m}{1 - \left[\frac{1}{2SP} \sum_{m=1}^{NC} \Gamma_m \right]^2} \quad (19)$$

Equation (18) is derived from the definition of C_L and the basic relationship between circulation and the lifting force. Equation (19) can be derived by writing $\tan \Delta\alpha$ as $\tan(\alpha_I - \alpha_E)$, expanding $\tan(\alpha_I - \alpha_E)$ into an equation with $\tan \alpha_I$ and $\tan \alpha_E$, and then using the velocity triangle to express $\tan \alpha_I$ and $\tan \alpha_E$ in terms of Γ_m , $\sin \alpha$ and $\cos \alpha$.

2.3 Transformation of Circular Cascades

Up to this point, the analysis has dealt with linear infinite cascades, and is applicable to the blade sections of axial flow machines. For radial flow machines, it is also desirable to be able to analyze inlet guide vanes of turbines and diffuser vanes of pumps and compressors. In the preceding analysis, linear cascades are handled by the concept that Equations (2) and (3) were derived for the velocities induced by infinite arrays of sources and vortices of constant spacing SP . The flow field thus repeats every spacing SP and satisfying boundary conditions on one airfoil of a cascade automatically satisfies the boundary conditions on all the airfoils of that cascade.

A circular cascade can also be considered an infinite cascade; because we can continually observe points around the cascade, and if all the vanes are identical the flow pattern repeats after each vane. If all the vanes are different, the vanes can be considered as several infinite cascades with the flow pattern repeating after each revolution.

Rather than deriving equations similar to Equations (2) and (3) for circular cascades, the approach taken was to mathematically transform the circular cascades into linear cascades and then use the existing analysis. Since the flow is being analyzed using potential theory, a conformal transformation can be used and the same governing equations and boundary conditions will apply in both the real and transformed planes. The transformation used, is performed by the following equations and is illustrated in Figure 3:

$$y = \theta \quad (20)$$

$$x = \ln r \quad (21)$$

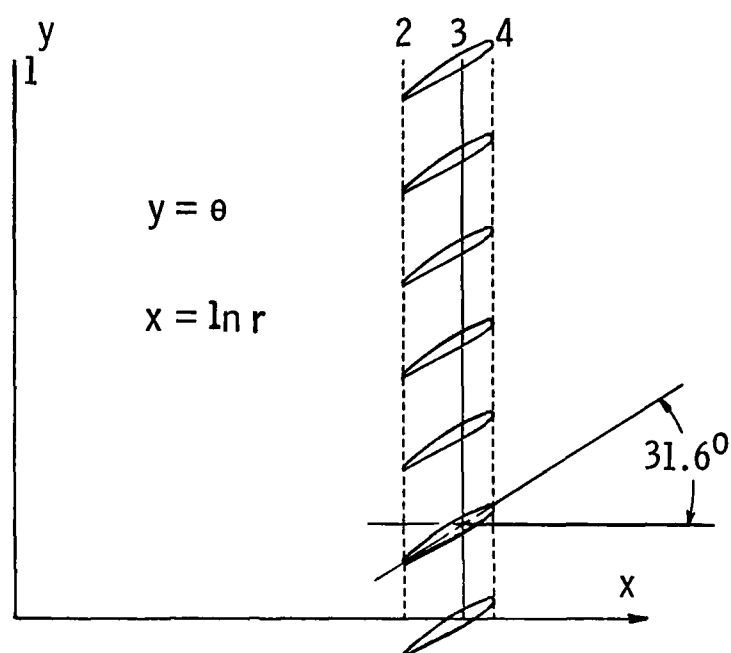
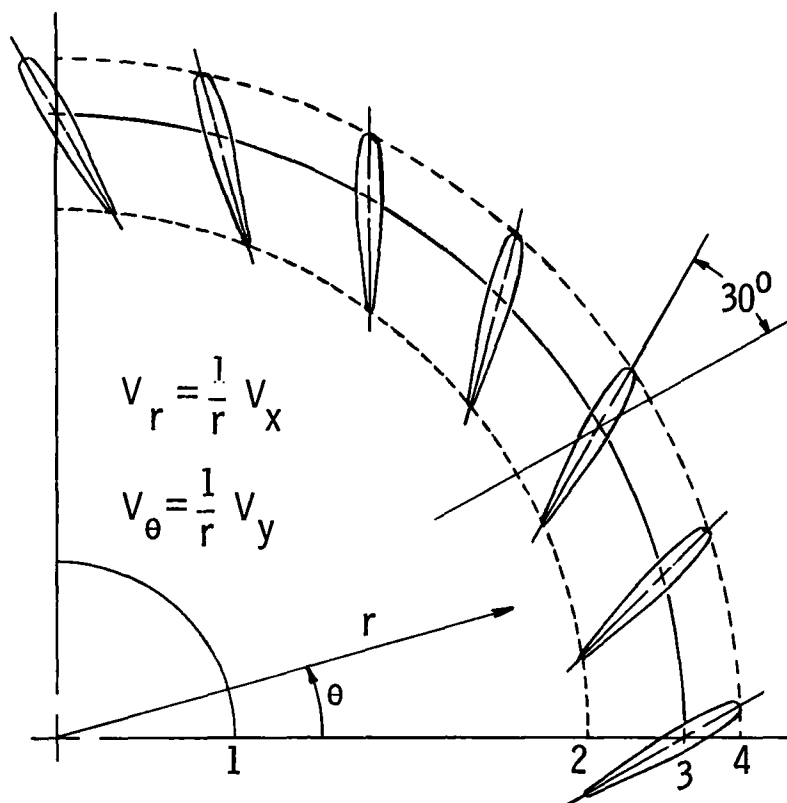


Figure 3. Schematic of a Circular Cascade (Above) Which is Transformed into the Linear Cascade (Below)

In addition to the equations for transforming the circular cascade geometry into a linear cascade, relationships are also required for relating the velocities in the two planes. From the velocities, the pressure field can then be calculated. The necessary relationships can be derived by first defining a stream function ψ in the r - θ plane such that:

$$V_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta} \quad (22)$$

and

$$V_\theta = - \frac{\partial \psi}{\partial r} \quad (23)$$

It is also necessary to define the stream function in the x - y plane such that:

$$V_x = \frac{\partial \psi}{\partial y} \quad (24)$$

$$V_y = - \frac{\partial \psi}{\partial x} \quad (25)$$

From the chain rule we know that

$$\frac{\partial \psi}{\partial \theta} = \frac{\partial \psi}{\partial y} \frac{\partial y}{\partial \theta} \quad (26)$$

and

$$\frac{\partial \psi}{\partial r} = \frac{\partial \psi}{\partial x} \frac{\partial x}{\partial r} \quad (27)$$

Differentiating Equations (20) and (21) we find that $\frac{\partial y}{\partial \theta} = 1$ and $\frac{\partial x}{\partial r} = \frac{1}{r}$. Substituting these relations and Equations (22) through (25) into Equations (26) and (27) yields the desired relationship between the velocity components in the real and transformed planes.

$$V_r = \frac{1}{r} V_x \quad (28)$$

$$V_\theta = \frac{1}{r} V_y \quad (29)$$

By combining the velocity components into the resultant velocity a similar equation relating the resultant velocities is obtained:

$$v_{r\theta} = \sqrt{v_r^2 + v_\theta^2} = \sqrt{\left(\frac{1}{r} v_x\right)^2 + \left(\frac{1}{r} v_y\right)^2} = \frac{1}{r} \sqrt{v_x^2 + v_y^2} = \frac{1}{r} v_{xy} \quad (30)$$

III. PROGRAM DESCRIPTION

The purpose of this section is to describe the key parameters and variables found in the program input and output, discuss the types of cascades which can be analyzed, and explain the structure of the program itself. The input format required to run the program is not presented in this section, but is presented in the users guide in Appendix A. In part one of this section, the program parameters and variables are defined and their application to the different types of cascades are discussed. The parameters used to nondimensionalize the data are also described. In part two, the restrictions which apply to the three basic types of cascades are presented, and ways for circumventing these restrictions to enable some unusual cascade geometries to be analyzed are explained. In the final part of this section, the structure of the program and the functions of each of the major subroutines are described.

3.1 Definition of Key Input and Output Variables

A convenient place to start the discussion of input and output variables is with the angles which describe the flow and the orientation of the airfoils in the cascades. The flow angles and the stagger angle ζ for a linear cascade are illustrated in Figure 2. In terms of the velocity components, the inlet angle α_I and the exit angle α_E can be defined as:

$$\alpha_I = \tan^{-1} \left(\frac{v_{yI}}{v_x} \right) \quad (31)$$

$$\alpha_E = \tan^{-1} \left(\frac{v_{yE}}{v_x} \right) \quad (32)$$

For a circular cascade, the definitions are similar with the x and y components of velocity replaced by the r and θ components, respectively. Thus, for a circular cascade with the radial flow direction inward or outward, the inlet and exit flow angles are defined as:

$$\alpha_I = \tan^{-1} \left(\frac{v_{\theta I}}{v_r} \right) \quad (33)$$

$$\alpha_E = \tan^{-1} \left(\frac{v_{\theta E}}{v_r} \right) \quad (34)$$

Equations (31) through (34) yield the sign which is consistent with the chosen convention. All flow angles for both linear and circular cascades are positive in the counterclockwise direction. The stagger angle for circular cascades is also measured in a counterclockwise direction, while the stagger angle for linear cascades is the only angle measured in a clockwise direction. The opposite sign convention was chosen for the linear cascade stagger angle to be consistent with the original Douglas Neumann program. For a typical cascade configuration as in Figure 2, the chosen convention also yields a positive lift corresponding to a positive stagger angle when the cascade is subjected to an axial inlet flow.

Two other angles which are sometimes used as input to the Douglas Neumann program are the flow turning angle $\Delta\alpha$ and the cascade mean flow angle α . These angles are defined as follows:

$$\Delta\alpha = \alpha_I - \alpha_E \quad (35)$$

$$\alpha = \tan^{-1} \left(\frac{\tan \alpha_I + \tan \alpha_E}{2} \right) \quad (36)$$

Tan α_I and tan α_E in Equation (36) can be expressed in terms of the x and y velocity components for a linear cascade. Making these substitutions would show that α is the angle of the average velocity vector through the cascades, and thus α has physical significance for linear cascades. For circular cascades, the physical significance of α is not as clear. Since the magnitudes of the inlet and exit velocities for circular cascades depend on the radius, no unique average velocity can be defined without specifying the radii. The inlet and exit velocities can each be converted to equivalent velocities at a common radius by applying conservation of mass ($rV_r = \text{constant}$) and conservation of angular momentum ($rV_\theta = \text{constant}$). The angle of the average of these equivalent velocities is α . At the common radius V_{r_I} is equal to V_{r_E} , just as V_{x_I} is equal to V_{x_E} for a linear cascade.

The angles α , α_I , α_E and $\Delta\alpha$ describe the overall characteristics of the flow through the cascade. However, the bulk of the output from the program is the velocity and pressure on the airfoil surfaces and at specified off-body points in the flow field. The velocities and pressures are both presented in nondimensional form, thus, the normalizing velocity for the various types of cascades should be discussed. For linear cascades, the magnitude U of the average velocity vector V_{avg} is used for non-dimensionalizing velocity and pressure. The vector V_{avg} is illustrated in Figure 2. From the figure it can be seen that:

$$U = \frac{V_x}{\cos \alpha} = V_I \frac{\cos \alpha_I}{\cos \alpha} = V_E \frac{\cos \alpha_E}{\cos \alpha} \quad (37)$$

For a circular cascade, an average velocity vector is difficult to define, therefore, a normalizing velocity with more physical significance was selected. With uniform flow, the volume flow rate Q for a radial flow system is $V_r 2\pi rW$. Even though the flow is not uniform through a cascade, the relationship for Q can be used to define a meaningful reference velocity by first writing:

$$V_r r = Q/2\pi W = \text{constant} \quad (38)$$

Since $V_r r$ is equal to a constant, selecting a reference radius provides a reference velocity which can be used to nondimensionalize the data for circular cascades. The quantity, W , is the width of the radial channel.

$$v_{r_{\text{ref}}} = \frac{1}{r_{\text{ref}}} \frac{Q}{2\pi W} = \frac{\text{constant}}{r_{\text{ref}}} \quad (39)$$

The value r_{ref} is given as input to the program and was selected as the radius which positions the airfoils in a cascade. The users guide in Appendix A will further illustrate r_{ref} .

Equation (39) defines the reference velocity for a circular cascade, but it is recalled that the flow field solution is originally obtained in a transformed plane where the reference velocity is U . A relationship between U in the transformed plane and $V_{r_{\text{ref}}}$ is therefore required in the program. Substituting r_{ref} for r and $V_{r_{\text{ref}}}$ for V_r in Equation (28) yields:

$$v_{r_{\text{ref}}} = \frac{1}{r_{\text{ref}}} v_x \quad (40)$$

From Equation (37) it is known that $V_x = U \cos \alpha$, thus:

$$V_{r_{ref}} = \frac{1}{r_{ref}} U \cos \alpha \quad (41)$$

Equation (30), which transforms a velocity in the x-y plane to a velocity in the r- θ plane, can now be written in nondimensional form as:

$$\frac{V_{r\theta}}{V_{r_{ref}}} = \frac{r_{ref}}{r} \frac{V_{xy}}{U \cos \alpha} \quad (42)$$

Equations similar to Equation (42) can be written for the velocity components V_r and V_θ by dividing Equations (28) and (29) by Equation (41).

All velocities in the output of the current version of the Douglas Neumann program are nondimensionalized by $V_{r_{ref}}$ and U for circular or linear cascades, respectively. The pressures on the body surfaces and at off-body points are also nondimensionalized by $V_{r_{ref}}$ and U . The pressure coefficient C_p is defined as:

$$C_p = 1 - \left(\frac{V_\ell}{V_{ref}} \right)^2 = \frac{P_{s\ell} - P_{s_{ref}}}{\frac{1}{2} \rho V_{ref}^2} \quad (43)$$

where $V_{ref} = U$ for linear cascades

$V_{ref} = V_{r_{ref}}$ for circular cascades

As seen from Equation (43), the pressure coefficient can be calculated from the local velocity V_ℓ at the point under consideration, but C_p also represents the difference between the local static pressure and a reference static pressure $P_{s_{ref}}$. The reference static pressure is defined as follows:

$$P_{s_{ref}} = P_T - \frac{1}{2} \rho V_{ref}^2 \quad (44)$$

where $V_{ref} = U$ or $V_{r_{ref}}$

P_T = total pressure, which is constant throughout the potential flow field

The remaining parameters to be defined again involve overall cascade properties rather than local pressures or velocities. For linear cascades, a lift coefficient C_L , x and y force coefficients C_{F_x} and C_{F_y} , and a moment coefficient C_M need to be defined. The lift coefficient C_L represents the total lift of all the cascades; and if it is not an input parameter, the program calculates C_L from the total circulation of the one or more cascades. If only one cascade is being analyzed, C_L is printed at the beginning of each page along with the other coefficients. However, for a multi-cascade analysis, C_L is printed at the end of the output of all the cascades so that it is not interpreted as the lift coefficient of a particular cascade.

The circulation on a cascade induces an upwash and downwash on the flow relative to the average flow through the cascade. For this reason, the lift force is perpendicular to the average velocity vector V_{avg} . C_L is defined by the following equation and is calculated in the program by the subsequent equation:

$$C_L = \frac{F_L}{\frac{1}{2}\rho U^2 cW} \quad (45)$$

where F_L = sum of the lift forces for one airfoil of each cascade.

F_L is perpendicular to V_{avg}

c = chord = 1

W = span, ($W = 1$ so that C_L represents the lift per unit span)

$$C_L = \frac{2}{Uc} \sum_{m=1}^{NC} \Gamma_m \quad (46)$$

where Γ_m = circulation on an airfoil in cascade m

NC = number of cascades

c = chord = 1

The coefficients C_{F_x} , C_{F_y} and C_M are obtained in a different manner than C_L , because they are computed by integrating the pressure distributions around the airfoils. For a multicascade analysis, coefficients are computed for each cascade. C_{F_x} and C_{F_y} nondimensionally represent the forces on the airfoils in the x and y directions, respectively, in the cascade global coordinate system. The moment represented by C_M is computed about a point specified in the input. These coefficients are defined as:

$$C_{F_x} = \frac{F_x}{\frac{1}{2}\rho U^2 c W} \quad (47)$$

$$C_{F_y} = \frac{F_y}{\frac{1}{2}\rho U^2 c W} \quad (48)$$

$$C_M = \frac{M}{\frac{1}{2}\rho U^2 c^2 W} \quad (49)$$

where F_x = force on the airfoil in the global x direction

F_y = force on the airfoil in the global y direction

M = moment about a point specified in the input (a positive moment is counterclockwise)

c = chord of the airfoils in the cascade for which the coefficients are calculated

C_L is related to C_{F_x} and C_{F_y} through the mean flow angle α . If there is more than one cascade, the force coefficients can be summed and used to compute C_L .

$$C_L = \sum_{m=1}^{NC} C_{F_{y_m}} c \cos \alpha - \sum_{m=1}^{NC} C_{F_{x_m}} c \sin \alpha \quad (50)$$

Equation (50) is not used in the program, but it is provided to explain the relationship between C_{F_x} , C_{F_y} and C_L . If C_L is hand calculated from Equation (50), the resulting value should only differ from the computer value by the small error incurred when integrating the pressure distributions for C_{F_x} and C_{F_y} . The chord c is required in Equation (50) for multiple cascade analyses where all the airfoils may not have $c=1$.

For circular cascades, C_L is not computed because an average velocity vector is not easily defined and C_L would have little meaning. Two force coefficients and a moment coefficient are computed for each circular cascade by integrating the pressure distributions. The following equations define these coefficients and complete the definition and discussion of the input and output variables.

$$C_{F_r} = \frac{F_r}{\frac{1}{2} \rho V_{r_{ref}}^2 cW} \quad (51)$$

$$C_{F_\theta} = \frac{F_\theta}{\frac{1}{2} \rho V_{r_{ref}}^2 cW} \quad (52)$$

$$C_M = \frac{m}{\frac{1}{2} \rho V_{r_{ref}}^2 c^2 W} \quad (53)$$

where F_r = force on the airfoil in the r direction

F_θ = force on the airfoil in the θ direction

M = moment about a point specified in the input (a positive
moment is counterclockwise)

3.2 Cascade Modeling

There are three basic types of cascades which the current version of the Douglas Neumann cascade program can handle. The program can analyze linear cascades, circular cascades with the radial flow inward, and circular cascades with the radial flow outward. In addition to single cascade analyses, multiple cascades can be analyzed as a system. For each of the various types of cascade analyses, the following restrictions apply to the cascades:

1. All the bodies of a particular cascade must be equally spaced.
2. All the bodies of a particular cascade must be identical.
3. For an analysis with multiple cascades, all the cascades must have the same spacing between the bodies.
4. No cascade can move relative to another cascade.

The last restriction is the only restriction which cannot be overcome through cascade modeling. Under no circumstance can a cascade move relative to another cascade because this would constitute an unsteady flow field.

For simple cascade geometries, no modeling is involved in the program input. The user simply inputs to the program the airfoil geometry and cascade parameters such as spacing, stagger angle, inlet flow angle, etc. However, for unusual cascade geometries where one of the first three restrictions is violated, a cascade can often be modeled as several cascades to overcome the restrictions.

Two examples will be used to illustrate cascade modeling. For the first example, the cascade of circular cylinders shown in Figure 4 has unequal spacing between the cylinders and thus violates the first restriction. In this case, although the distances from one cylinder to the next cylinders above and below are not equal, it is seen that the cascade pattern repeats every spacing SP. The single cascade of unequally spaced cylinders can thus be modeled as two cascades. Both cascades will have the same spacing between the cylinders, with the second cascade offset from the first cascade by the distance ADDY shown in the figure. The numbers on the cylinders in Figure 4 identify the cylinders belonging to the two cascades.

Although in the first example the cylinders of cascade one and two have the same diameter, the two cascades could have different size or differently shaped bodies. Thus, the same procedure can be used to model cascades with bodies of different geometries. The process of representing differently shaped bodies or equally spaced bodies as different cascades could be carried to an extreme where a random cluster of differently shaped two-dimensional objects could be analyzed. Each object could be represented as a cascade, and the spacing could be made very large such that the effect of other members of the same cascade would be negligible. This extreme situation may require a very large number of points to represent the different body geometries and, thus, may not be practical. However, the extreme case does illustrate the flexibility of the program made available by the ability of the program to analyze multiple cascades. It should be possible to model most cascades of engineering significance with a reasonably number of cascades which will not violate any of the program restrictions.

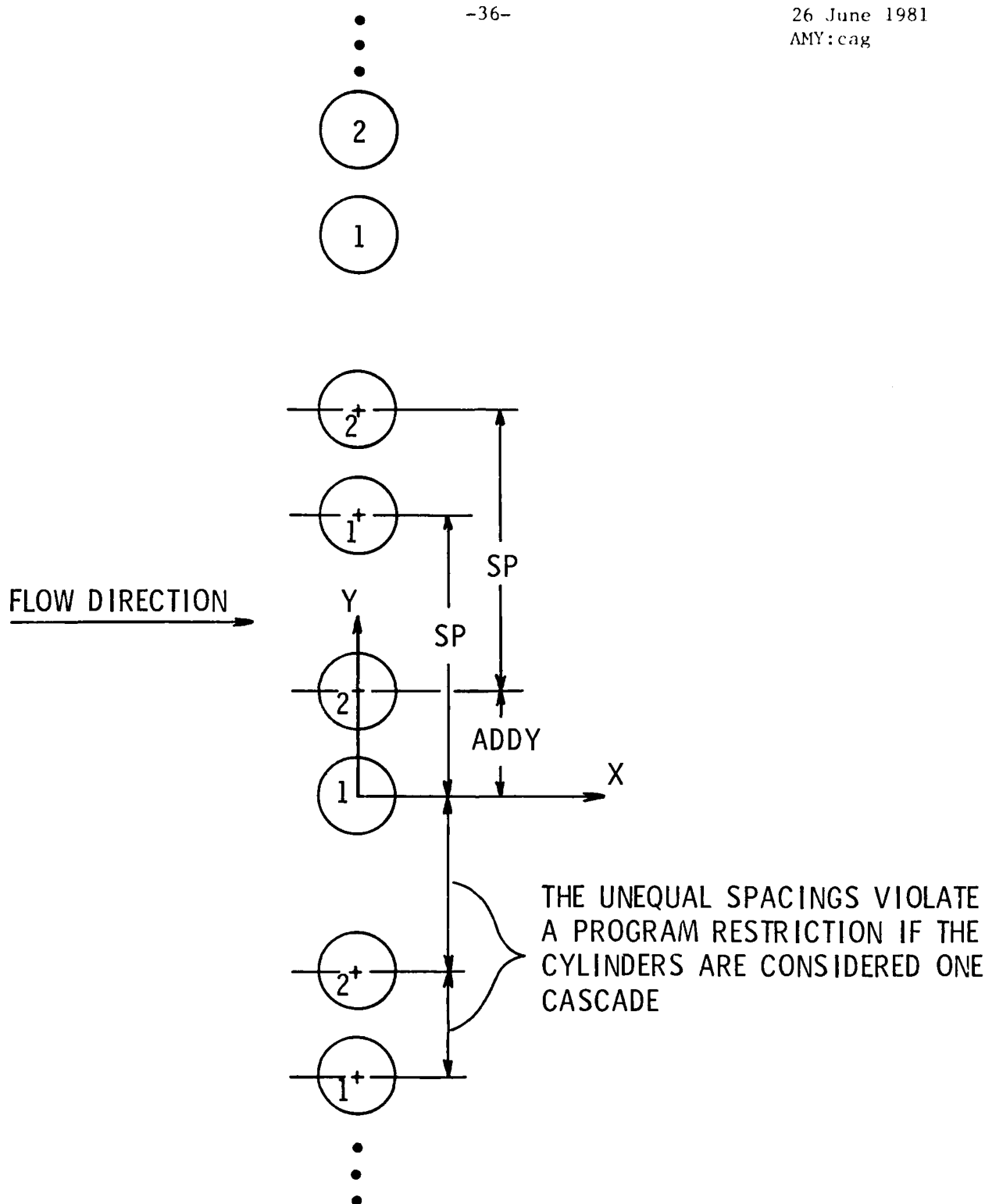


Figure 4. Illustration of How a Cascade of Cylinders With Unequal Spacing Can be Modeled as Two Cascades of Equal Spacing S_p , Offset by the Distance $ADDY$

The second example of cascade modeling illustrates an analysis of a cascade with apparently differently shaped airfoils. Figure 5 is a schematic of stay vanes and wicket gates of a hydraulic turbine. It is noticed from the figure that the stay vanes have five different geometries. Over most of the circumference the stay vane geometry is constant with either stay vane A or stay vane B being employed. In the regions where the stay vane geometry does not vary, the flow can easily be analyzed with the current version of the Douglas Neumann program. The wicket geometry and the appropriate stay vane geometry are input to the program, neglecting the other stay vane geometries which are remote from the area of interest.

Neglecting the variation of the stay vane geometry, the wicket gates and stay vanes represent two cascades which do not violate any of the program restrictions. Suppose, however, that it is desirable to analyze the flow in the region of stay vane D, where it is seen from Figure 5 that there are five different stay vane geometries side by side. Since there is no symmetry in the distribution of stay vane geometries, to analyze the entire flow field including the five different stay vane geometries would require 48 cascades. Each wicket gate and stay vane would need to be represented as a separate cascade with the spacing $SP=360^\circ$. The number of points required to represent 48 cascades certainly makes this approach unrealistic.

A more realistic approach to analyze the flow near vane D is to only consider the two vanes on both sides of vane D, and after these vanes allow the pattern to repeat. In this way, the effects of the two nearest vanes in both directions is included, but the effects of the true vane shapes beyond the first two vanes are neglected in the model. This

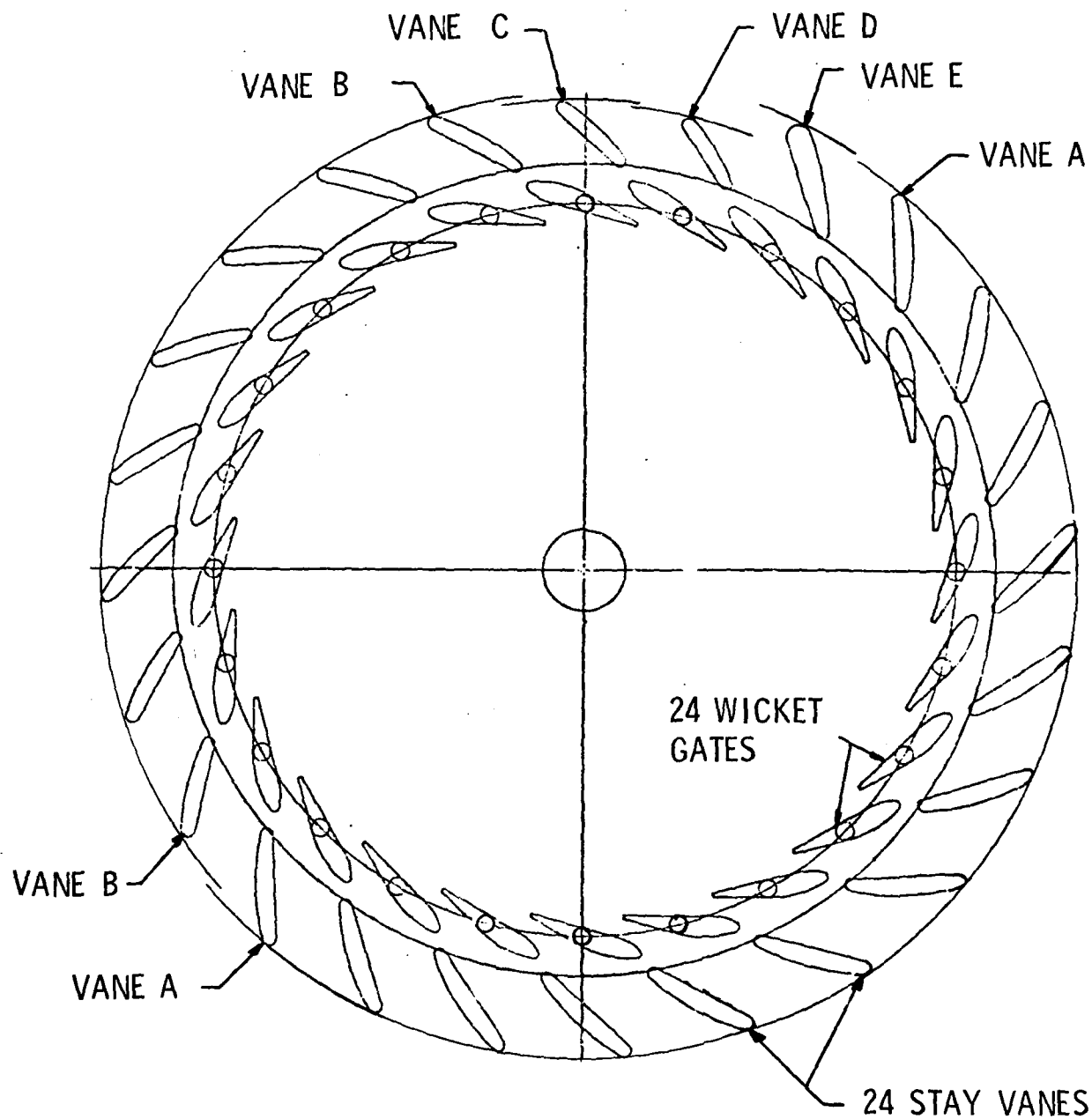


Figure 5. Wicket Gates and Stay Vanes of a Hydraulic Turbine

approach is illustrated in Figure 6. In order that the wicket gate spacing be the same as the stay vane spacing, the wicket gates corresponding to the five stay vanes must also be modeled as different cascades. The need to represent the wicket gates as five cascades, even though they have the same geometry, is also apparent from the fact that they are each downstream of a different stay vane geometry and will each have a different flow solution. The flow in the region of stay vane D can therefore be modeled with ten cascades, the maximum number the program is currently set up to handle.

In Figure 6 the ten cascades representing the wicket gates and stay vanes near vane D are numbered to identify the members of the various cascades. The original vane shapes are also identified by the letters. It is seen that each cascade has the same spacing SP and after the spacing SP the pattern repeats. The relative position of the various cascades is specified by the radius and the angles $ADDY_1$, $ADDY_2$, etc.

It is important to remember when modeling different vane shapes, as in the preceding example, that the vane of primary concern should be kept in the middle of the group. This will minimize errors caused by not having the true vane shape a few vanes from the place of concern. For example, the model shown in Figure 6 should not be used to analyze the flow near vane C, but instead, a new model with two true vanes on both sides of vane C should be employed.

Note: A circular cascade becomes an infinite linear cascade in the transformed plane because each succeeding revolution around the circular cascade will generate more of the same airfoils. In the example where there were 24 stay vanes, the true pattern would repeat every 24 vanes. However, the stay vanes were modeled as 5 cascades which cannot be evenly

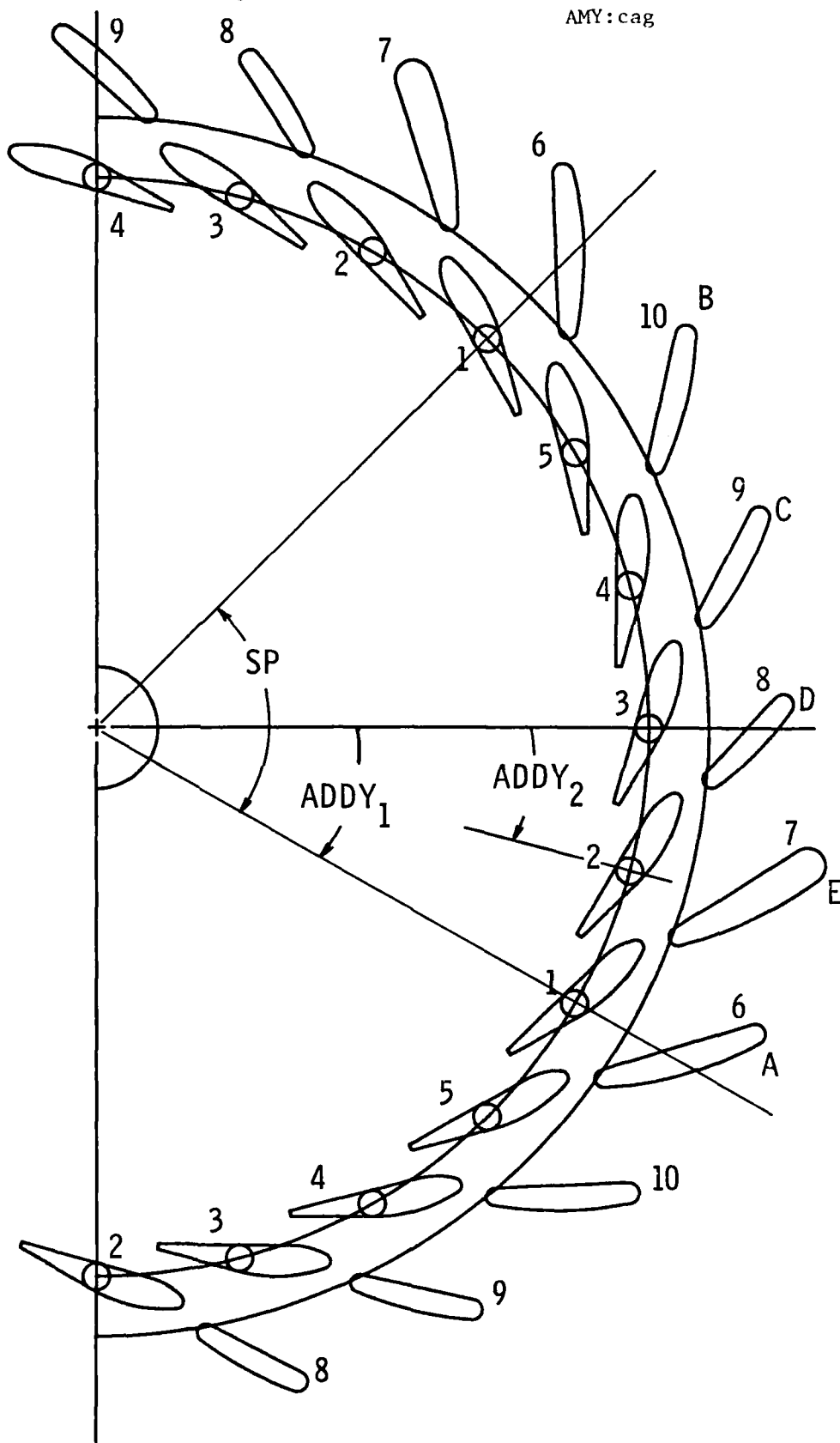


Figure 6. Illustration of One Possible Way to Model the Wicket Gates and Stay Vanes in the Region of Stay Vane D of Figure 5

divided into 24. Therefore, the pattern will not repeat every 24 vanes as it should. This fact may be somewhat disturbing because transforming the linear cascade back to the circular cascade will result in several different vane shapes occupying the same location. In the transformed plane, where the flow field solution is originally obtained, the airfoils which will occupy the same location in the real plane are 24 vanes apart. Thus, they have essentially no effect on each other, and the fact that the pattern does not repeat every 24 vanes is inconsequential. The author has not verified this conclusion with computer runs of the program, therefore, it may be wise for the user to do so if the number of cascades used in modeling a circular cascade cannot be divided evenly into the total number of vanes.

3.3 Structure of the Program

The current version of the Douglas Neumann cascade program consists of a main program and five major subroutines. The main program's only function is to call the appropriate major subroutines. Each of the major subroutines will be described in the following paragraphs. Besides the CALL statements for the subroutines, the main program also contains two logical IF statements. One IF statement selects SUBROUTINE PART3 or SUBROUTINE PART4 for solving the set of simultaneous equations depending on the size of the problem. The other IF statement directs the program to go to SUBROUTINE PART5 following SUBROUTINE PART1 for cases where the basic solutions of the previous case can be used for the present case. The main program will continue to process cases until an input code directs the program to be terminated. If cases of similar geometry are run where only the flow angle or lift coefficient is changed, the same basic solutions can be used for each case. Subsequent to the first solution

when the basic solutions are obtained, the program only needs to solve the combination equations. This feature will save a considerable amount of computation time when several solutions with the same cascade geometry but different flow conditions are required:

SUBROUTINE PART1:

The major purpose of SUBROUTINE PART1 is to read the input data and do any of the preliminary calculations which are necessary before the set of simultaneous equations can be set up and solved. One of the first cards to be read is a control card specifying the type of cascade and giving other run control information. Cascade parameters are read next, followed by the airfoil coordinate data. The airfoil coordinate data is usually read from cards. However, if the airfoil geometry is the same as the previous case, the data can be read from a file. For runs where a particular airfoil geometry is repeated for modeling purposes, as in the stay vane - wicket gate example in the preceding subsection, SUBROUTINE PART1 will generate the airfoil coordinate data for the similar airfoils following the input of the first airfoil data. After the coordinate data have been input, the data are stored in a file before the required manipulations begin.

For circular cascades, SUBROUTINE PART1 reorders the coordinate data so that in the transformed plane the data will have the same order as data for linear cascades. The necessity of reordering the data is described in more detail in Appendix C. For circular cascades with the radial flow direction inward, the coordinate data are put in reverse order. If the radial flow direction is outward, the starting point of the data is shifted to the trailing edge of the airfoil for the current flow direction.

SUBROUTINE PART1 next scales the coordinate data by the value of the chord given for each cascade. The coordinate data are then converted from local to global coordinates, which involves translating and rotating the coordinates so the airfoils have the proper location and orientation in the flow field. For circular cascades, the transformation is carried out at the same time the coordinates are converted to global coordinates. Coordinates of the element midpoints are determined and the sine and cosine of the angle of the elements are calculated. The airfoil coordinates are put in complex form to be used in the complex arithmetic involved in setting up the A matrix in Equation (11).

The last section of SUBROUTINE PART1 reads in or generates the coordinates of off-body points where the flow is to be determined. The coordinates of each off-body point can simply be read in from a card. However, in many cases it is desirable to locate a prescribed number of off-body points equally spaced across a section of the flow at a constant x or r value. Under these circumstances, the program can generate the coordinates of the off-body points after being given the initial data for a set of points. By supplying the program with the coordinates of the first point of the set, the number of off-body points in the set, and the increment Δy or $\Delta \theta$ by which to locate the remaining points, the program will generate the required coordinates. Any number of sets of off-body points can be used, provided the total number of off-body points does not exceed the array dimensions. For circular cascades, the off-body points are transformed to the linear cascade by SUBROUTINE PART1.

SUBROUTINE PART2:

In SUBROUTINE PART2, the set of simultaneous equations which are solved for the source distributions on the airfoils are set up. Setting up the equations consists of calculating the values of the A matrix and determining the onset flows in Equation (11). Complex arithmetic is used to calculate the A matrix, such that the real and imaginary parts yield the coefficients for the flow induced both tangent and normal to the airfoil surface. The coefficients representing the flow induced normal to the surface are the values in the A matrix. The coefficients for the flow induced tangent to the surface are stored and later used to calculate the velocity on the airfoil surfaces after the source strengths are known. The sum of the real parts and the sum of the imaginary parts of the coefficients for each row of the matrix are also calculated in SUBROUTINE PART2. The sum of the real parts is the term $\sum_{k=n_1}^{n_2} A'_{jk}$ in Equation (12) and provides the onset flows induced normal to the element surfaces by circulation. The sums of the imaginary parts provide the velocities induced tangent to the elements by circulation and are required to calculate the final velocities at the element midpoints.

SUBROUTINE PART3 and SUBROUTINE PART4:

The set of simultaneous equations for each basic solution are solved in either SUBROUTINE PART3 or SUBROUTINE PART4 depending on the size of the problem. SUBROUTINE PART4 reads the coefficients and onsets flows calculated in SUBROUTINE PART2 from files and stores the values in arrays. SUBROUTINE MIS1 is then called, which solves the equations by matrix reduction. Since SUBROUTINE PART4 and MIS1 require the entire A matrix and onset flow matrices to be in the computer core at once, the use of these subroutines is restricted by the size of the matrices which can be accommodated in the core.

There are currently two versions of SUBROUTINE PART3 which will solve the equations when it is not practical or possible to have the entire A matrix in the computer core. The original Douglas Neumann cascade program employed an iterative technique to solve the equations for large problems. This iterative technique is found in one of the two current versions of SUBROUTINE PART3. A disadvantage of the iterative approach is that the complete set of equations must be read from auxiliary storage for each iteration. Reading data from auxiliary storage is one of the slower operations of computer processing and, thus, may greatly increase the computation time.

To reduce the computation time for large problems, a new solution procedure was developed at ARL/PSU during the current rewriting of the Douglas Neumann program. The new procedure found in the second version of SUBROUTINE PART3 reduces the A matrix by operating on blocks of rows of the A matrix in the core. The procedure is designed to take advantage of whatever core space is available and it continually optimizes the equation storage by increasing the number of equations in the core as the number of nonzero terms in each equation decreases. Comparative computer runs solving the same problem using the three solution procedures show that the new procedure requires the least amount of computer time.

The three available solution procedures found in SUBROUTINE PART4 and the two versions of SUBROUTINE PART3 are described in more detail in Appendix B. Discussions of the storage requirements and the major advantages and disadvantages of each procedure are included. Appendix B also describes the program dimension statements so that the user can choose a solution procedure and modify the array dimensions to match the program for a particular problem to the computer system.

SUBROUTINE PART5:

The last major subroutine is SUBROUTINE PART5. Its first function is to set up and solve the combination equations. The solution of the combination equations enables the basic solutions to be combined such that the cascade operating condition and the Kutta condition are satisfied. As discussed in Section 2.2, the combination equations are based on a ΔV across the trailing edge which represents each of the basic solution's violation of the Kutta condition. The beginning of SUBROUTINE PART5 calculates the required ΔV 's.

Following the determination of the ΔV 's there are four sections of the subroutine which solve the combination equations depending on which of the four possible input parameters were given. If α or α_I is the given cascade operating parameter, the combination equations are solved by matrix reduction. If C_L or $\Delta\alpha$ is given, the program iterates to determine the value of α the flow must have to yield the given C_L or $\Delta\alpha$. The solution of the combination equations produces the value of the circulation for each cascade.

Following the solution of the combination equations, SUBROUTINE PART5 calculates the velocity at the midpoint of each element. This is done by adding the onset velocities and the velocities induced by all the elements for each of the basic solutions. When adding solutions, the velocities of the appropriate basic solutions are scaled by $\sin \alpha$, $\cos \alpha$ or the value of the circulation for each cascade. For circular cascades, the velocities are next transformed back to the real plane and the data are reordered to correspond to the data original input order. The airfoil pressure distribution is calculated from the velocities and then is integrated to yield

force and moment coefficients. The results are printed and the velocity calculation process repeats for the other cascades of a multi-cascade analysis. The last section of SUBROUTINE PART5 calculates the velocity components and the pressure at each of the specified off-body calculation points.

IV. PROGRAM VERIFICATION

4.1 Methods of Verification

Every computer program should be checked to assure that it is calculating what it was intended to calculate. Since the Douglas Neumann program yields an approximate solution to a potential flow problem, the obvious means of verifying the program is to analyze the flow through a cascade for which an exact potential flow solution exists. In this section, results from the current rewritten version of the Douglas Neumann program are compared to exact solutions for various types of cascades. The exact solutions are obtained by using conformal transformations to convert the potential flow around a single cylinder to the flow around airfoils in infinite cascades. Besides the comparisons with exact solutions, one additional comparison is made between the results from the current version of the program and results from the original version which were presented in Reference [1]. This comparison was made to assure the current version of the program can analyze multiple cascades of lifting and nonlifting bodies as did the original program.

It is also important to know how well a computer program predicts real phenomena or experimental data. However, the degree to which calculated results match experimental data is not necessarily a check

on the proper functioning of a computer program. Instead, comparisons with experimental data are checks on how well the simplifying assumptions and governing equations represent the real situation. Comparisons of results from the Douglas Neumann program and experimental data have been made in other reports but will not be made here. In Reference [1], measured pressure distributions for linear cascades are compared to calculated results from the program. Measured and calculated flow angles for a circular cascade with the radial flow direction inward are compared in Reference [2].

With the current version of the Douglas Neumann program having the additional capability of analyzing circular cascades, several new solutions as well as the original solutions for linear cascades required verification. The types of cascades for which the solutions required verification were:

1. A linear cascade.
2. A circular cascade with the radial flow inward.
3. A circular cascade with the radial flow outward.
4. Multiple cascades of each of the first three types.
5. Multiple cascades consisting of both lifting and nonlifting bodies.

Solutions for the first three types of cascades were verified by comparing them with their corresponding exact analytical solution. For multiple cascades of each type, a single cascade was modeled as several cascades, thus the same exact solution could be used to verify both single and multiple cascades. As previously mentioned, the fifth type of cascade solution was verified by comparing the results from the current version of the program with the results presented in Figure 5 of Reference [1].

The cascade parameters C_L (C_L is a parameter for linear cascades only), α_I , α , and $\Delta\alpha$ were all known from the exact solutions for the various types of cascades. Thus, each of the cascade parameters could be used as input to the program to check the various input options. Using each of the different cascade parameters as input, the other parameters were calculated by the program. In this way, the cascade parameters calculated by the program were verified. As required, the calculated pressure distributions on the airfoils were nearly identical regardless of the input parameter. Velocity components at points off the bodies were also obtained from the exact analytical solutions and were used to check the off-body data calculated by the program.

4.2 Exact Analytical Cascade Solutions

Exact potential flow solutions for linear and circular cascades can be obtained from the flow around a circular cylinder by employing a series of conformal transformations. The transformations used to obtain a linear cascade are the same as those found in References [1] and [3]. An additional transformation was then employed to transform the linear cascade to a circular cascade.

Flow around a circular cylinder is generated by locating a pair of sources and vortices at points B' and B'' and a pair of sinks and vortices at points A' and A'' shown in the S -plane of Figure 7. The locations of the source, sink and vortex pairs determines the position of the cylinder. The position of the cylinder in turn determines the final shape of the airfoils after the transformation. To form a cylinder, the sink and vortex at point A' must mirror the sink and vortex at A'' , and similarly the singularities at B' and B'' must do the same. The two points of each pair lie on the same radial line and the radial distance

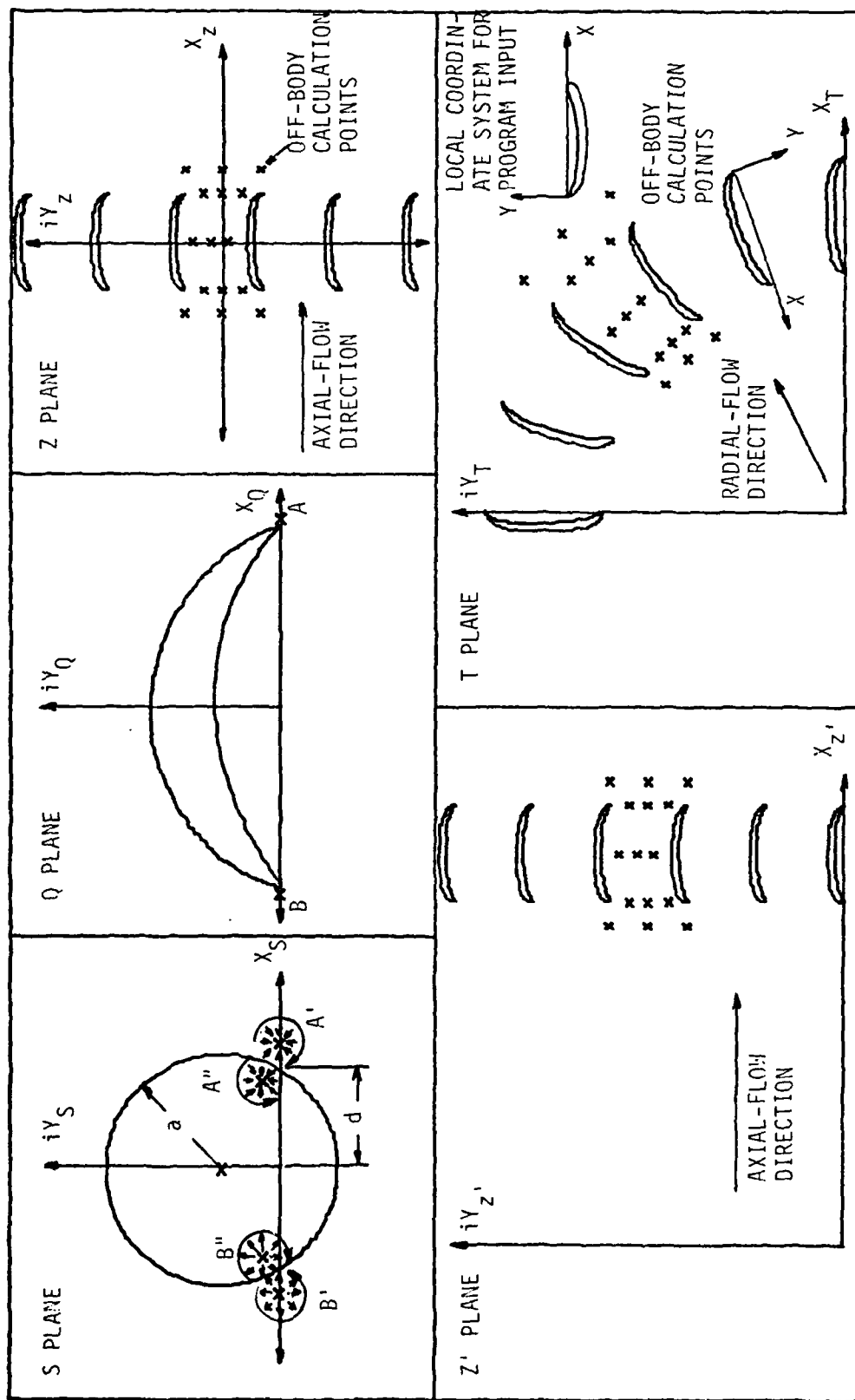


Figure 7. Illustration of the Transformation of a Cylinder in the S-Plane to the Two-Dimensional Linear Cascade in the Z-Plane and the Addition Transformation of the Linear Cascade to the Circular Cascade in the T-Plane. The Radial Flow is Outward in the T-Plane

from the center of the cylinder to the inner point is the inverse of the radial distance to the outer point (i.e. if A' is r from the center, A'' is $1/r$ from the center). To satisfy continuity, the source and the sink have the same strength. Together, the source, sink and vortex strengths determine the inlet flow angle and must also be specified so as to satisfy the Kutta condition. With this brief background, the following parameters and coordinates provide the necessary information for the cylinder used:

a = radius of the cylinder = 1.0
 d = distance to the point where the cylinder intersects
 the x_s axis (this point becomes the T.E. of the airfoils)
 d = .8460254

Point	Coordinates
Center of the Circle	(-0.02, 0.5)
A'	(1.06605, 0.0)
A''	(0.739739, 0.150228)
B'	(-1.088605, 0.0)
B''	(-0.787722, 0.140783)

Two sequential transformations are used to map the cylinder in the complex S-plane into a cascade of airfoils. The first transformation maps the cylinder into a single airfoil shape in the complex Q-plane as illustrated in Figure 7. This transformation is performed by the following equation:

$$\frac{Q - rd}{Q + rd} = \left(\frac{S - d}{S + d} \right)^r \quad (54)$$

where r = trailing-edge-angle constant = 1.85

Q = complex coordinates in the Q -plane

S = complex coordinates in the S -plane

The second transformation maps the airfoil shape in the complex Q -plane into a cascade of airfoils in the complex Z -plane. This transformation is:

$$Z = \ln \left(\frac{Q - B}{Q - A} \right) \quad (55)$$

A and B are the complex coordinates of the points in the Q -plane corresponding to the points A' and B' . Points A and B are illustrated in Figure 7. The numerical value of A and B can be found by plugging the given coordinates for A' and B' into Equation (54), respectively.

The linear infinite cascade of airfoils resulting from the second transformation is illustrated in the Z -plane of Figure 7. This is the cascade which was analyzed with the current version of the Douglas Neumann program to verify the program functions properly. The cascade was analyzed as a single cascade and was also modeled and analyzed as multiple cascades. As an example of a multiple cascade model, the numbers above the airfoils in Figure 7 identify the members of each cascade for a two cascade model.

To obtain a circular cascade for which an exact potential flow solution is possible, a third transformation is applied to the linear cascade. Before the third transformation is applied, however, the linear cascade is modified slightly. For the case illustrated in Figure 7, where the circular cascade has the radial flow direction inward, the linear cascade is translated and the signs of the x_z , y_z coordinates

are changed. These changes, resulting in the cascade illustrated in the Z' -plane of Figure 7, are necessary to obtain the desired airfoil orientation and flow direction for the circular cascade in the T -plane. In Figure 8, the transformations leading to a circular cascade with the radial flow direction outward are illustrated. Up through the Z -plane, the transformations are identical to those in Figure 7. In the Z' -plane of Figure 8, however, it is seen that the cascade only needs to be translated to the $+x_z$ direction to yield the desired airfoil orientation and flow direction in the T -plane.

The transformation used to go from the Z' -plane to the T -plane is the same transformation used in the program, except that for the exact solutions it is used in complex form. To obtain the new form, Equations (20) and (21) are first repeated here in terms of the nomenclature of Figures 7 and 8.

$$y_{z'} = \theta_T \quad (56)$$

$$x_{z'} = \ln r_T \text{ or } r_T = e^{x_{z'}} \quad (57)$$

It is necessary to express the cartesian coordinates of the T -plane in terms of r_T and θ_T .

$$x_T = r_T \cos \theta_T \quad (58)$$

$$y_T = r_T \sin \theta_T \quad (59)$$

Applying the identities of Equations (56) and (57) to Equations (58) and (59) yields:

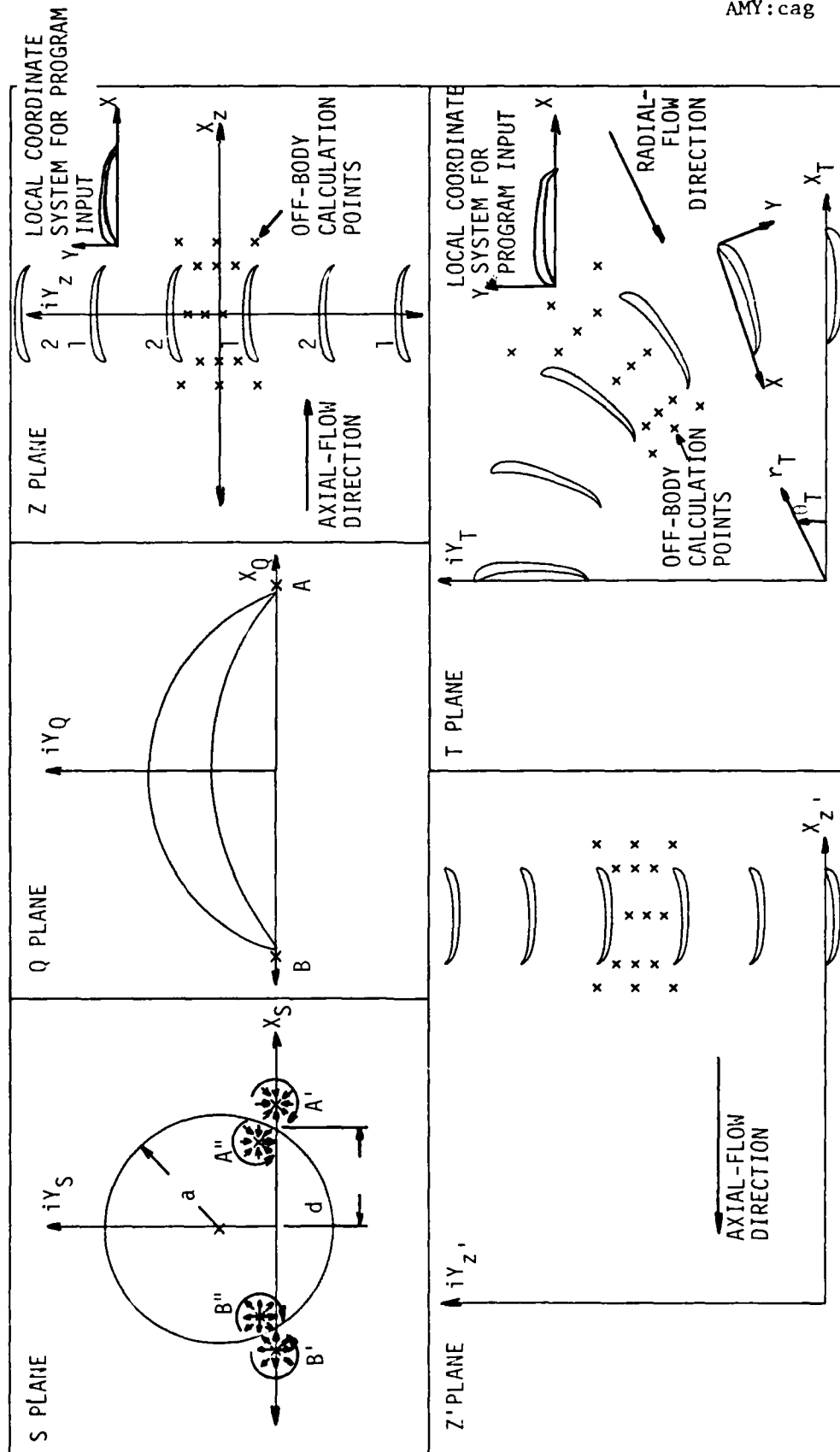


Figure 8. Illustration of the Transformations Which Yield a Circular Cascade in the T-plane With the Radial Flow Inward

$$x_T = e^{x_{z'}} \cos y_{z'}, \quad (60)$$

$$y_T = e^{x_{z'}} \sin y_{z'}, \quad (61)$$

Expressing x_T and y_T as a complex number yields the new form of the transformation.

$$T = x_T + i y_T = e^{x_{z'}} (\cos y_{z'} + i \sin y_{z'})$$

$$= e^{x_{z'}} e^{iy_{z'}} = e^{x_{z'} + iy_{z'}}$$

$$T = e^{Z'} \quad (62)$$

This new expression of the transformation in complex form facilitates the calculation of the complex velocities in the exact solution of the flow field for the circular cascades.

Starting with coordinates of points on the surface of the cylinder in the S-plane the preceding transformations define the cascade geometries for which an exact potential flow solution is obtained. The transformations also provide the local coordinates of points defining the airfoil shapes, which are required as input to the Douglas Neumann program. To obtain the exact flow field solution, however, it is necessary to apply the transformations to the complex potential function which represents the flow around the cylinder in the S-plane. The complex potential function $F(S)$ is derived by adding the contribution of each singularity

$$F(S) = \underbrace{-E \ln(S - A')}_{\text{sink}} + \underbrace{iG \ln(S - A')}_{\text{vortex}} - \underbrace{E \ln(S - A'')}_{\text{sink}} - \underbrace{iG \ln(S - A'')}_{\text{vortex}}$$

(63)

$$+ \underbrace{E \ln(S - B')}_{\text{source}} - \underbrace{iH \ln(S - B')}_{\text{vortex}} + \underbrace{E \ln(S - B'')}_{\text{source}} + \underbrace{iH \ln(S - B'')}_{\text{vortex}}$$

where E = source and sink strength

G = vortex strength at A' and A''

H = vortex strength at B' and B''

Equation (63) can be written in simplified form as:

$$F(S) = K_1 \ln(S - A') + \bar{K}_1 \ln(S - A'') + K_2 \ln(S - B') + \bar{K}_2 \ln(S - B'') \quad (64)$$

where $K_1 = -E + iG$

\bar{K}_1 = complex conjugate of K_1

$K_2 = E - iH$

\bar{K}_2 = complex conjugate of K_2

Of greater significance than the complex potential function is its derivative $W(S)$, which is the complex velocity in the S-plane.

$$W(S) = \frac{dF(S)}{dS} = \frac{K_1}{(S - A')} + \frac{\bar{K}_1}{(S - A'')} + \frac{K_2}{(S - B')} + \frac{\bar{K}_2}{(S - B'')} \quad (65)$$

To get the complex velocity in other planes, the chain rule is applied.

For the Z plane, $W(Z)$ is obtained as follows:

$$W(Z) = \frac{dF(Z)}{dZ} = \frac{dF(S)}{dS} \frac{dS}{dQ} \frac{dQ}{dZ} \quad (66)$$

or for the T-plane:

$$W(T) = \frac{dF(T)}{dT} = \frac{dF(S)}{dS} \frac{dS}{dQ} \frac{dQ}{dZ} \frac{dZ}{dZ'} \frac{dZ'}{dT} \quad (67)$$

Taking the required derivatives of the equations representing the various transformations yields:

$$\frac{dS}{dQ} = \frac{S^2 - d^2}{Q^2 - r^2 d^2} \quad (68)$$

$$\frac{dQ}{dZ} = \frac{Q^2 - Q(A + B) + AB}{A - B} \quad (69)$$

$$\frac{dZ}{dZ'} = \text{a constant which depends on which radial cascade is desired} \quad (70)$$

$$\frac{dZ'}{dT} = e^{-Z'} \quad (71)$$

The preceding equations complete all the equations which are required to calculate the velocity at any point in the various planes. It should be noticed that to calculate the derivatives in Equations (68), (69) and (71), the complex coordinates of the point under consideration are required in the S, Q and Z' planes. Thus, to calculate the velocity at a point in the T-plane, for example, the corresponding points in the other planes must first be determined using the transformations of Equations (54), (55) and (62). After the coordinates of these points are known, the velocity at the original point in the S-plane is calculated using Equation (65). This complex velocity is then multiplied by the derivatives calculated from Equation (68) through (71) yielding the desired velocity at the point in the T-plane.

Before concluding this section on calculating an exact potential flow solution for an analytically derived cascade geometry, the constants K_1 , \bar{K}_1 , K_2 and \bar{K}_2 remain to be specified. These constants are chosen so as to yield the desired inlet or average flow condition and also to satisfy the Kutta condition. Since \bar{K}_1 and \bar{K}_2 are complex conjugate of K_1 and K_2 , respectively, it is only necessary to determine K_1 and K_2 . The desired flow conditions are known or specified far upstream and far downstream of the cascade in the Z-plane. Thus, it is desirable to express K_1 in terms of $W(Z)$ as x_z approaches plus infinity and K_2 in terms of $W(Z)$ as x_z approaches negative infinity. The points A and B go to plus infinity and minus infinity, respectively, during the transformation from the Q-plane to the Z-plane. Thus, taking the limit of $W(Z)$ as S approaches A' and Q approaches A yields the limit of $W(Z)$ as x_z approaches plus infinity. From this limit the following equation results:

$$W(Z) = K_1 = V_x - iV_y, \quad \text{at } x_z = +\infty. \quad (72)$$

Similarly, taking the limit of $W(Z)$ as S approaches B' and Q approaches B yields:

$$W(Z) = -K_2 = V_x - iV_y, \quad \text{at } x_z = -\infty. \quad (73)$$

The velocity components in Equations (72) and (73) can be expressed in terms of the magnitude of the average cascade velocity U , the cascade average flow angle α and the upwash and downwash velocity created by the circulation on the airfoils.

$$K_1 = U[\cos \alpha - i(\sin \alpha - V_{dn})] \quad (74)$$

$$K_2 = U[-\cos \alpha + i(\sin \alpha + V_{up})] \quad (75)$$

(V_{up} and V_{dn} are nondimensionalized by U)

In order for the velocities and coefficients calculated with the exact solution to match the data from the Douglas Neumann program, K_1 and K_2 are nondimensionalized by U yielding:

$$K_1 = \cos \alpha - i(\sin \alpha - V_{dn}) \quad (76)$$

$$K_2 = -\cos \alpha + i(\sin \alpha + V_{up}) \quad (77)$$

(Note: The equations for the exact solution in Reference [1] are

nondimensionalized by the magnitude of the inlet velocity V_I)

By definition, V_{up} is equal to V_{dn} . Thus, if α is specified, V_{up} and V_{dn} constitute one unknown which can be solved for by requiring the trailing edge of the airfoils to be a stagnation point. However, rather than arbitrarily selecting α , V_{up} and V_{dn} were expressed in terms of the lift coefficient C_L which was given the arbitrary value of 1.75 for the current check cases. With C_L specified, α must be determined such that the Kutta condition is satisfied. The expressions for K_1 and K_2 in terms of α and C_L are:

$$K_1 = \cos \alpha - i\left(\sin \alpha - \frac{C_L}{4SP}\right), \quad (78)$$

$$K_2 = -\cos \alpha + i\left(\sin \alpha + \frac{C_L}{4SP}\right), \quad (79)$$

where $SP = \text{dimensionless cascade spacing} = \frac{\text{cascade spacing}}{\text{chord}}$.

Inserting the preceding expressions for K_1 and K_2 along with their complex conjugates into Equation (65) provides the necessary equation for determining α . The complex velocity $W(S)$ must be zero at the point on

the cylinder where it intersects the x_s axis, since this point becomes the trailing edge of the airfoils in the Z-plane. Solving for α with a simple interval halving technique yields $\alpha = -0.86154$ degrees.

The determination of α completes the explanation of the exact analytical cascade solution. Important parameters for the cascade in the Z-plane are summarized below:

$$SP = 0.79675$$

$$C_L = 1.75$$

$$\alpha = -0.86154 \text{ degrees}$$

$$\alpha_I = 28.10800 \text{ degrees}$$

$$\alpha_E = -29.43192 \text{ degrees}$$

$$\Delta\alpha = 57.53992 \text{ degrees}$$

The values given above for the angles are also true for the circular cascades since angles are preserved by conformal transformations.

4.3 Comparisons of the Program Results With Exact Analytical Solutions

The first comparison of the program results with an exact analytical solution is made for the linear cascade illustrated in the Z-plane of Figure 7. The pressure distributions on these airfoils obtained from the cascade program and the exact solution are shown in Figure 9. It is seen from Figure 9 that the agreement in the data is excellent, as it should be if the program is functioning properly. Near the center portion of the airfoil, the differences between the two results are the largest. For the computer run, the points representing the airfoil were more closely spaced near the leading and trailing edges of the airfoil in order to adequately describe the airfoil shape. More points in the center region of the airfoil should improve the results in this area. Near the leading and trailing edges where a large number of points were

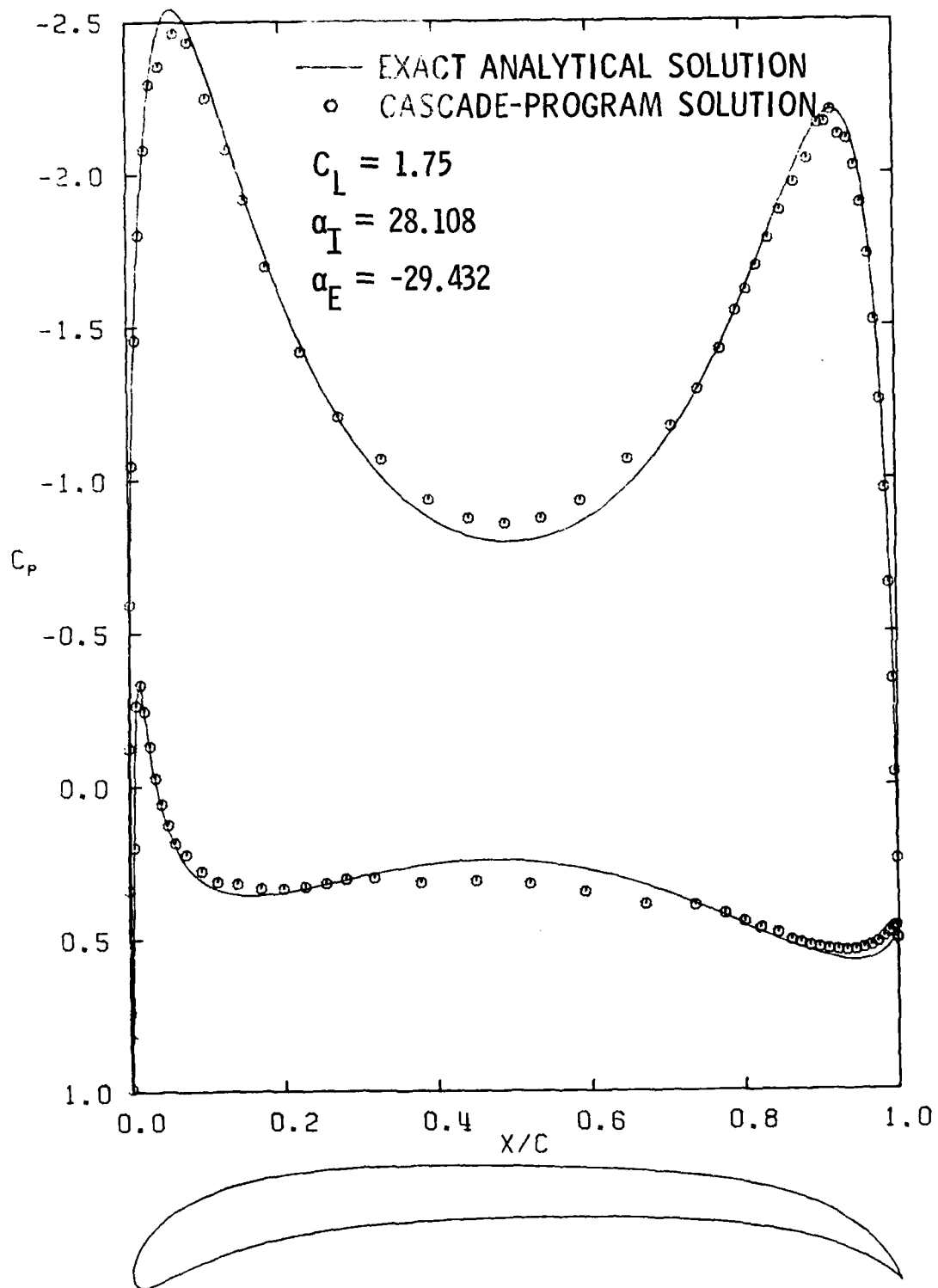


Figure 9. A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Analytically Derived Airfoil Shape in a Linear Cascade. The Cascade Geometry is Illustrated in the Z-Plane of Figure 7

26 June 1981
AMY:cag

used, the steep pressure gradients are very accurately calculated by the cascade program.

The cascade solution yielding the pressure distribution shown in Figure 9 was obtained by representing the airfoil with 99 points. As just stated, more points should yield a more accurate solution, but even with 99 points the results are probably within required engineering accuracy. Most airfoil shapes employed in engineering practice will not have the large amount of blade curvature present in the analytical profile. For this reason, 100 or less points will usually adequately represent any airfoil of engineering significance. A means of checking if a given number of points is sufficient to represent an airfoil is to change the number of points and determine if the solution is significantly altered. It is recalled that the technique employed in the Douglas Neumann program will yield results which approach the exact solution as the number of points approaches infinity.

In order to verify that the program functions properly for each of the various input options, several cases were run for each cascade with the different cascade operating parameters given as input. In addition, the options of running a case as a subcase of the previous case or directing the program to go directly to the combination equations were checked. Using the linear cascade as an example, C_L was the cascade operating parameter given as input for the first case. For the second case, α was the input parameter and the case was run as a subcase of the first case. For the third and fourth cases, $\Delta\alpha$ and α_I were the input parameters, respectively, and the program was directed to go directly to the combination equations. The pressure distributions calculated by the program for each of the four cases were found to be nearly identical. Comparisons

of the cascade parameters calculated by the program for the four cases along with the values obtained from the exact analytical solution are presented in Table 1.

TABLE 1

A Comparison of Cascade Parameters Obtained
From the Program for the Various Input Parameters

Case #	Input Parameter	Values Obtained From the Cascade Program				
		C_L	α	$\Delta\alpha = \alpha_I - \alpha_E$	α_I	α_E
1	C_L	—	-0.7712	57.5388	28.1768	-29.3620
2	α	1.7451	—	57.4040	28.0392	-29.3648
3	$\Delta\alpha$	1.7503	-0.7645	—	28.1871	-29.3618
4	α_I	1.7475	-0.8165	57.4714	—	-29.3634
Values From the Exact Solution		1.75	-0.8615	57.5399	28.1080	-29.4319

From the fact that the pressure distributions for the four cases were nearly identical and from the results shown in Table 1, it is apparent that the various input and running options in the program function properly.

The linear cascade shown in the Z-plane of Figure 7 will also be used for an example of the verification of the program results for off-body points. The locations of the off-body points are marked with "x's" in Figure 7. Table 2 lists the coordinates of the off-body points along with the velocity components at these points obtained from both the cascade program and the exact analytical solution. The values shown in the table verify that the program properly calculates the velocity components at off-body points.

TABLE 2

A Comparison of Velocity Components at Off-Body Points
Obtained From the Cascade Program and the Exact Analytical Solution

Off-Body Point Coordinates		Cascade Program Results		Exact Analytical Solution	
x/c	y/c	V_x/U	V_y/U	V_x/U	V_y/U
-0.7424	0.3984	0.9663	0.5518	0.9683	0.5502
-0.7424	0.0	1.0239	0.5195	1.0252	0.5180
-0.7424	-0.3984	0.9663	0.5518	0.9683	0.5502
-0.4924	0.1992	0.9756	0.3831	0.9762	0.3829
-0.4924	0.0	1.1106	0.4345	1.1110	0.4333
-0.4924	-0.1992	1.2289	0.6373	1.2285	0.6344
0.0	0.3011	0.9510	0.0088	0.9526	0.0086
0.0	0.1188	1.0827	-0.0028	1.0830	-0.0031
0.0	-0.0635	1.2305	-0.0133	1.2306	-0.0138
0.5077	0.1992	0.9427	-0.4307	0.9404	-0.4314
0.5077	0.0	1.0824	-0.4592	1.0809	-0.4601
0.5077	-0.1992	1.1879	-0.6175	1.1868	-0.6188
0.7577	0.3984	0.9806	-0.5822	0.9790	-0.5840
0.7577	0.0	1.0190	-0.5445	1.0174	-0.5461
0.7577	-0.3984	0.9806	-0.5822	0.9790	-0.5840

In Figure 10, the pressure distributions are presented for the airfoils in the circular cascade with the radial flow inward. This cascade is illustrated in the T-plane of Figure 7. As seen from Figure 10, the results from the exact analytical solution and the cascade program are nearly identical. Again, the differences in the results should be reduced by increasing the number of points.

It is interesting to compare the shape of the pressure distribution for the airfoils in the linear cascade with the pressure distribution for the airfoils in the circular cascade. From Figure 9, it is seen that the pressure distribution for the linear cascade has two nearly

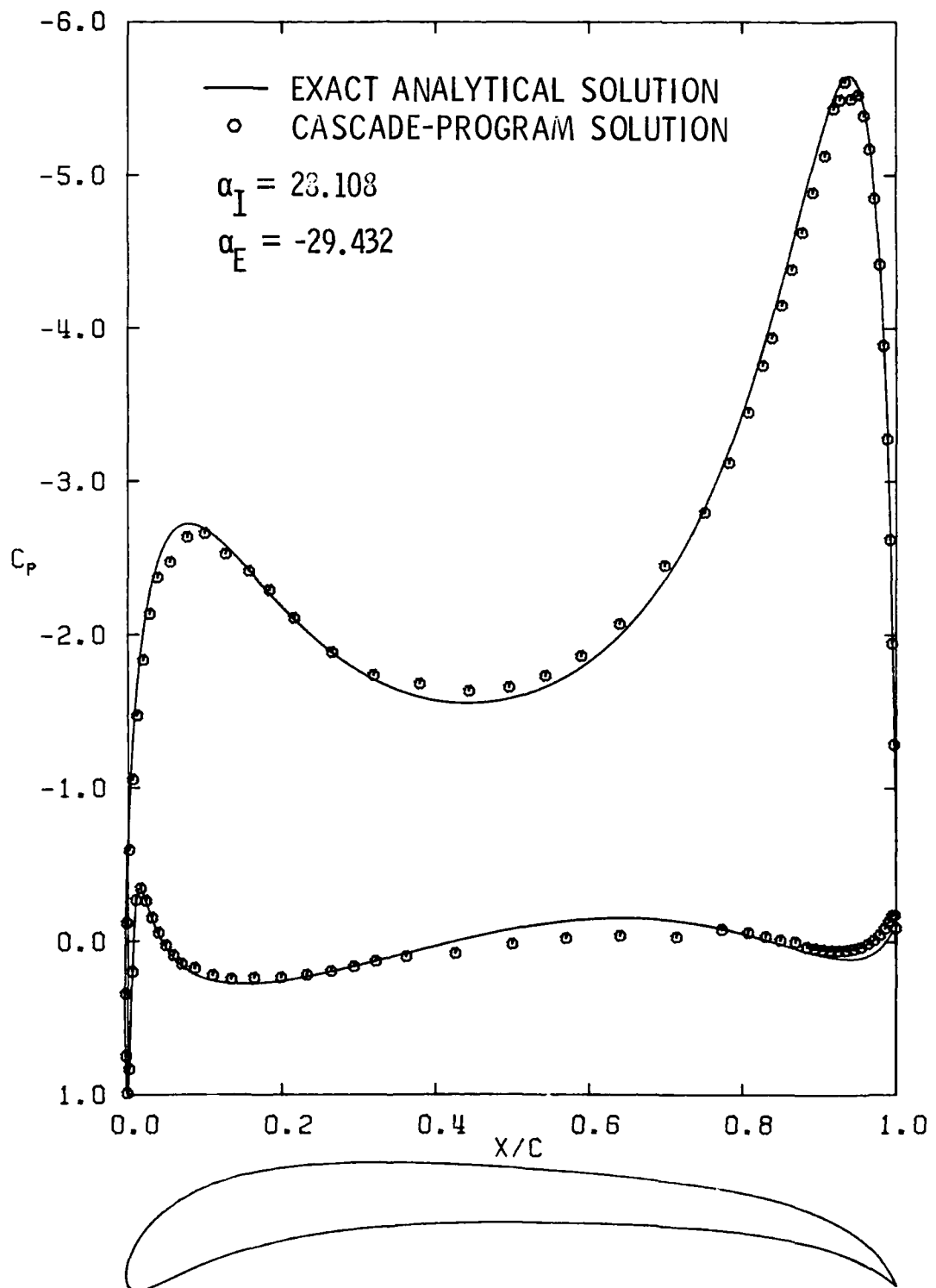


Figure 10. A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Circular Cascade Illustrated in the T-Plane of Figure 7

symmetrical peaks. The pressure distribution shown in Figure 10 for the circular cascade with the radial flow inward has one large peak and one small peak. The airfoil geometries in the two cascades are a little different due to the transformation from the Z-plane to the T-plane. However, the major effect causing the pressure distributions to have different shapes is the change in radius for the circular cascade. The trailing edge of the airfoil in the circular cascade is at a smaller radius than the leading edge. For this reason, the velocity near the trailing edge will be greater than the velocity near the leading edge, resulting in the trailing edge peak to be greater than the peak in the pressure distribution near the leading edge.

Figure 11 shows the comparison of the pressure distributions for the last type of cascade, the circular cascade with the radial flow outward. This cascade is illustrated in the T-plane of Figure 8. In Figure 8, notice the orientation of the airfoil in the local coordinate system used for the program input. Since the airfoil is inverted in the local coordinate system, the airfoil is also shown this way in Figure 11. In addition, the scale for C_p is inverted from the previous scales to match the airfoil. For the circular cascade with the radial flow outward, it is again seen that the pressure distributions computed with the exact analytical solution and the rewritten version of the Douglas Neumann program are nearly identical.

The pressure distribution shown in Figure 11 is similar to the pressure distribution in Figure 10, in that it has one large peak and one small peak. However, for the circular cascade with the radial flow outward, the large peak is near the leading edge, since the leading edge is at the smaller radius where the velocity is larger.

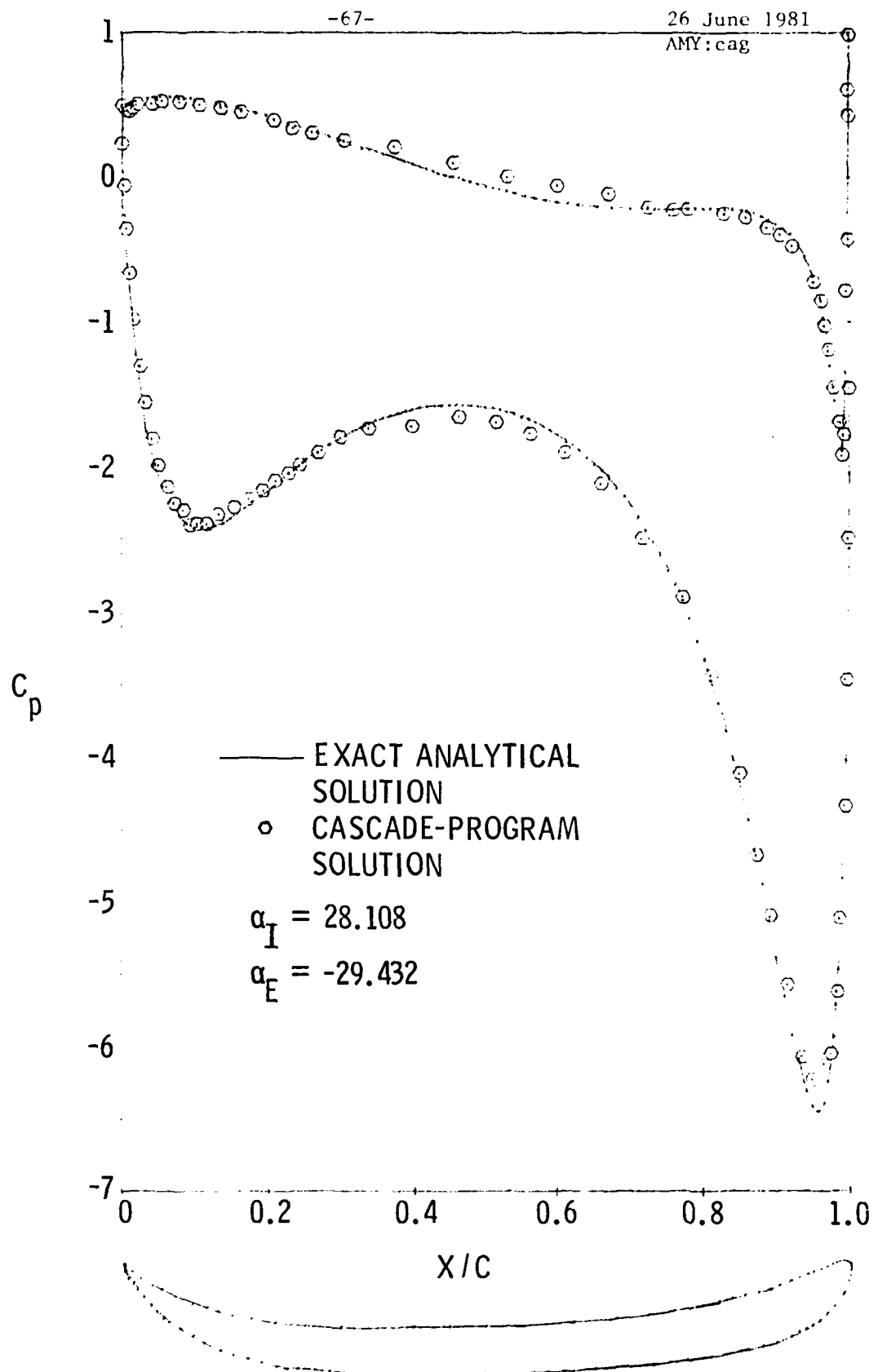


Figure 11. A Comparison of the Pressure Distributions Obtained From the Exact Solution and the Computer Program for the Circular Cascade Illustrated in the T-Plane of Figure 8

For both types of circular cascades, the various input options, program running options, and the off-body point results were all verified in the same manner as explained for the linear cascade. The three types of single cascades were then modeled as multiple cascades to verify the program results for multiple cascade analyses. The pressure distributions for the different cascades of a multiple cascade model were found to be nearly identical to each other and to the results for the single cascade. The flow angles and off-body point results were also found to be correct for multiple cascade analyses.

4.4 Verification of Results for Multiple Cascades of Lifting and Nonlifting Bodies

A special case of a multiple cascade analysis results when some cascades have lifting bodies and others have nonlifting bodies. To verify that the current version of the Douglas Neumann program could handle this special case, results from the current version of the program were compared to results published for the original program in Reference [1]. The cascades analyzed and the resulting pressure distributions are shown in Figure 12. The first and third cascades consist of cylinders with no circulation. The second cascade consists of cylinders which have circulation. In the computer program, the circulation resulted from specifying the rear stagnation point (trailing edge) to be at -30° on the cylinder. As seen from Figure 12, the pressure distributions from the two programs are nearly identical, which verifies the current version of the program can correctly analyze the special case of multiple cascades with lifting and nonlifting bodies. (Note: The first and third cascades do have lift forces on them due to induced effects of the second cascade. However, they do not produce lift on their own.)

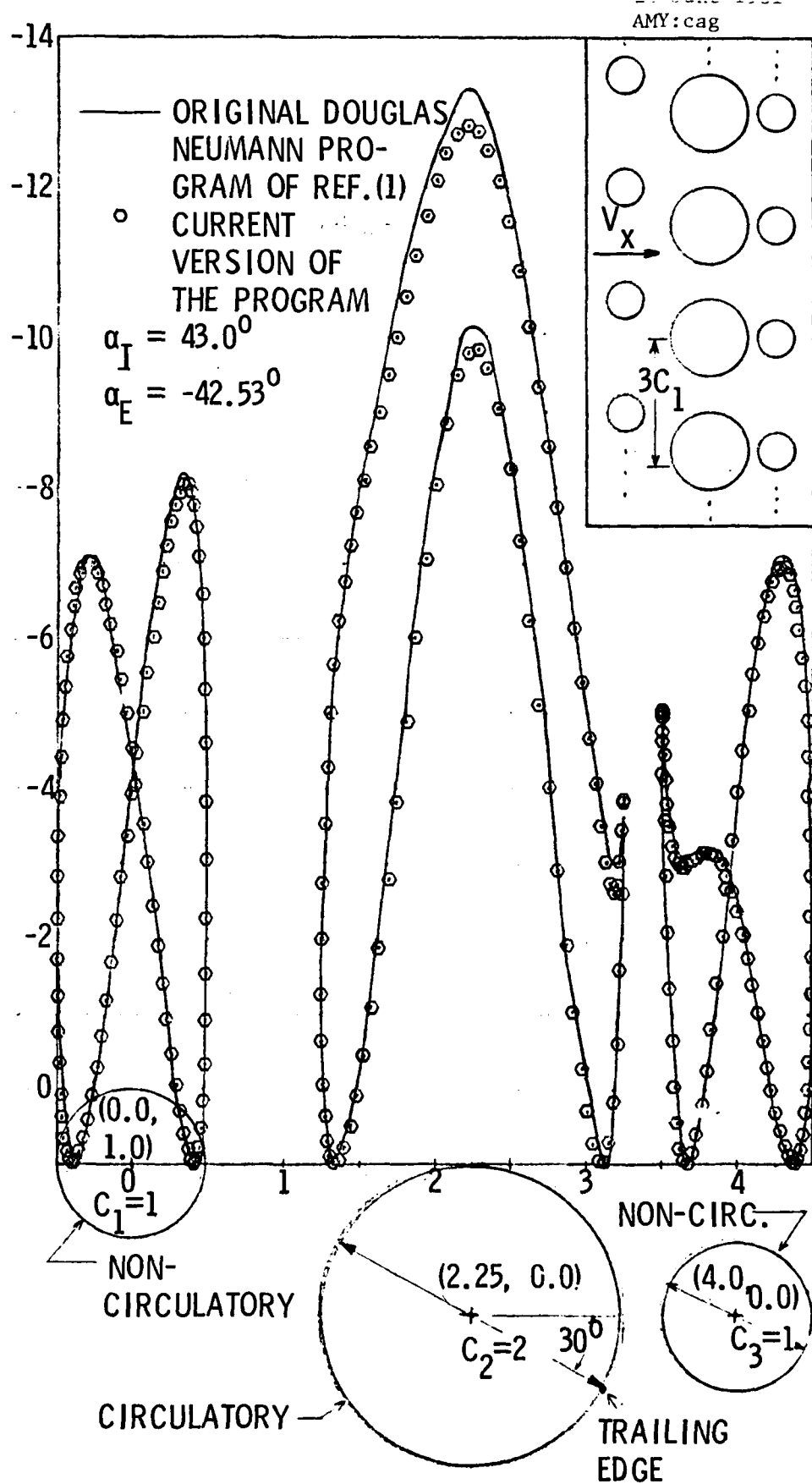


Figure 12. A Comparison of Pressure Distributions for Cascades of Lifting and Non-Lifting Bodies Obtained From the Original and the Current Version of the Douglas Neumann Cascade Program

REFERENCES

- [1] Giesing, J. P., "Extension of the Douglas Neumann Program to Problems of Lifting, Infinite Cascades," U.S. Department of Commerce, Report No. LB31653, AD605207, revised July 2, 1964.
- [2] Gearhart, W. S., Yocum, A. M., and Seybert, T. A., "Studies of a Method to Prevent Draft Tube Surge and the Analysis of Wicket Gate Flow and Forces," Prepared for the Bureau of Reclamation by the Applied Research Laboratory, Pennsylvania State University, Report No. REC-ERC-78-12, April 1979.
- [3] Argyris, J. H. and Mareczek, G., "Potential Flow Analysis by Finite Elements," Ingenieur-Archiv 41, 1972.

APPENDIX A

A Users Guide for Running the Program

A.1 Explanation of Program Input

The main body of this report has given the theory behind the program, has defined the input and output variables, and has discussed how various types of cascades can be modeled. A basic understanding of the preceding material should enable the program to be used and the results understood with little difficulty. This appendix will give detailed instructions on how to input the required data to the program. In the latter sections of this appendix, sample input and sample output are presented.

Before describing the program input, attention is called to Figures A1 and A2 which illustrate the geometric parameters involved in the input. Figure A1 illustrates two linear cascades and Figure A2 illustrates two circular cascades. Input for one cascade or more than two cascades follow the same format. With regard to circular cascades, it should be pointed out that except for two parameters, the input for circular cascades with the radial flow inward and the input for circular cascades with the radial flow outward are identical. Of the two parameters which differ, one tells the program the flow direction and the other specifies the location of the trailing edge when the radial flow direction is outward.

For each input card required by the program, the title of the card will be followed by the variables which appear on the card and their required format. After the list of variables, a description of each variable will be given. The following sequence of cards can be repeated several times in order to run several cases at once. The program will continue to process runs until a code on the second card tells the

program to terminate. The various options available for analyzing a special case of the previous run will be explained as the appropriate input variables are encountered.

Heading Card: HEDR FORMAT (20A4)

HEDR - A heading which describes the run and is read as alphanumeric characters. This heading is printed at the top of each page of output.

Control Card: NB, FLG02, FLG03, FLG04, FLG05, FLG06, FLG07, FLG08,
FLG09, FLG10, FLG11, FLG12 FORMAT (12I5)

NB - number of bodies or cascades

FLG02 - A nonzero integer if the flow is to be determined at points off of the bodies.

As stated in the main body of this report, α , $\Delta\alpha$, α_I , or C_L can be used in the combination equations for specifying the operating condition of the cascades. FLG03 through FLG06 specify which input parameter is used.

FLG03 - A nonzero integer if α is to be used in the combination equations.

FLG04 - A nonzero integer if $\Delta\alpha$ is to be used in the combination equations.

FLG05 - A nonzero integer if α_I is to be used in the combination equations.

FLG06 - A nonzero integer if C_L is to be used in the combination equations.

C_L cannot be used as input for circular cascades.

FLG07 - A nonzero integer if the matrices of influence coefficients (A and B) and the onset flows are to be printed. The A matrix represents the velocity components induced normal to the surface and the B matrix represents the velocity components induced tangent to the surface. A and B are large matrices and normally are not printed (FLG07 = 0).

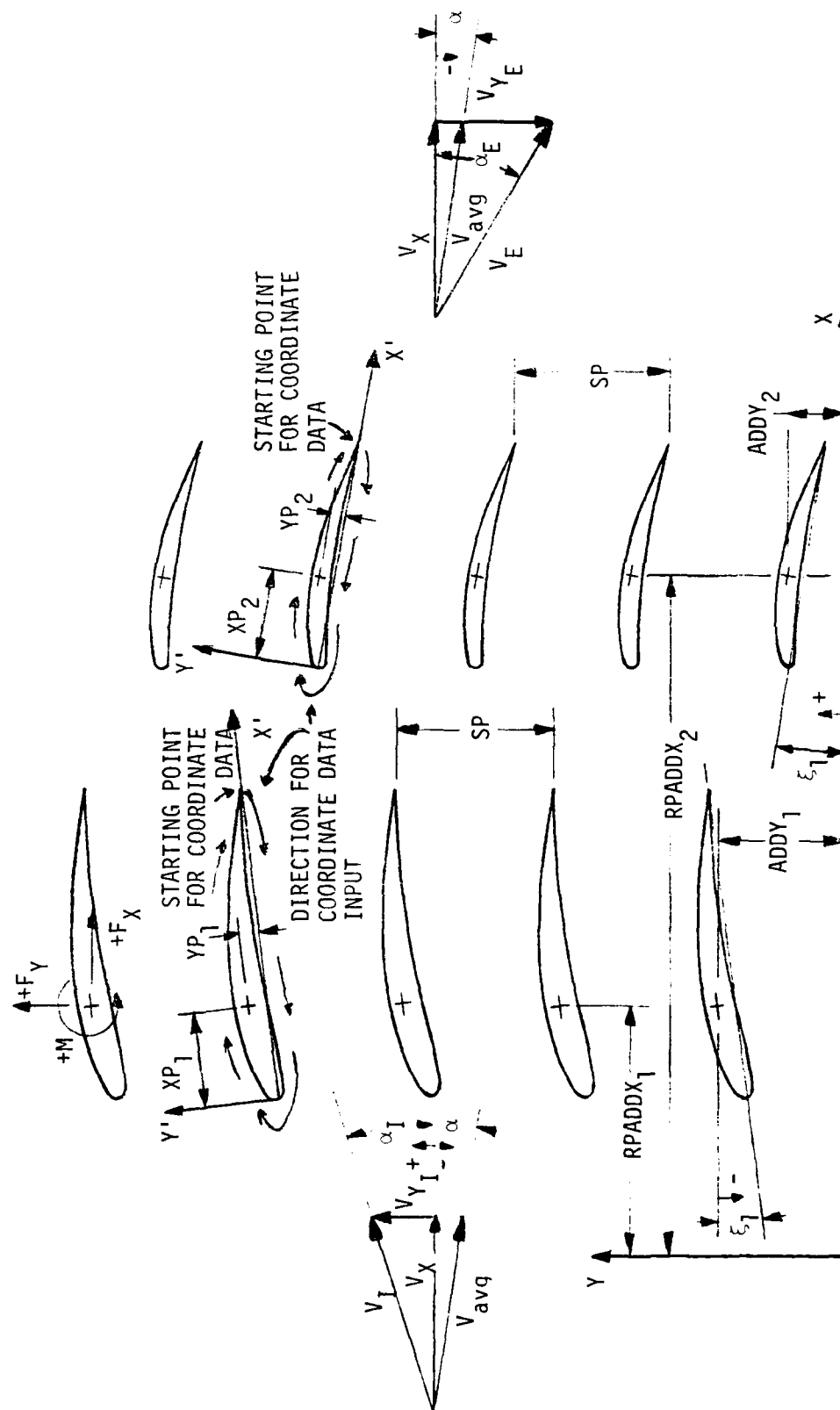


Figure A.1. Schematic of Two Linear Cascades Illustrating the Geometric Input Parameters for the Program

26 June 1981

AMY:cag

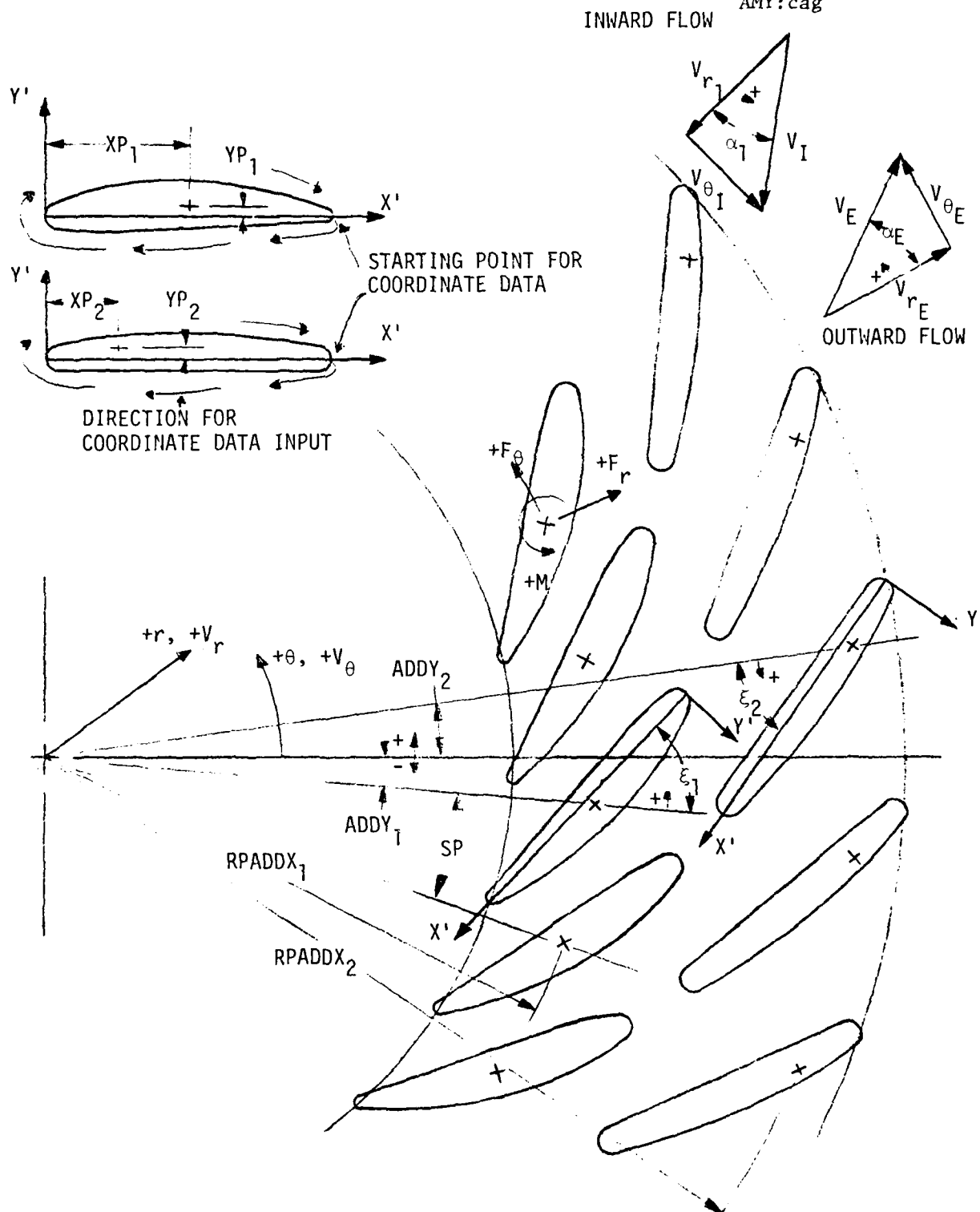


Figure A.2. Schematic of Two Circular Cascades Illustrating the Geometric Input Parameters for the Program. The Cascade Geometric Input are Independent of the Radial Flow Direction

- FLG08 - A nonzero integer if the program is to go directly to the combination equations using the basic solutions of the previous case. This option can be used only if the cascade geometry is identical to the previous case (i.e. only the flow angle or C_L is changed). Off-body point data input is independent of FLG08.
- FLG09 - A nonzero integer if the cascades under consideration are circular cascades.
- FLG10 - For circular cascades, FLG10 is a nonzero integer if the radial flow is outward and FLG10 is zero if the radial flow is inward.
- FLG11 - A nonzero integer if the first cascade geometry is to be repeated for a multiple cascade analysis. This option was included as an aid in inputting data when modeling the cascades requires one geometry to be repeated with equal spacing between each body (as with the wicket gates in the example illustrated in Figure 6).
- FLG12 - A nonzero integer if the program is to be terminated. The program will continue to try to process additional runs until terminated. Thus, terminating the program at the end of a run requires a heading card, which can be left blank, and a control card with FLG12 a nonzero integer.

Cascade Parameter Card: SP, CL, ALPHA, FALPHA, DALPHA, SUBKS FORMAT (5F10.0, I5)

SP - Cascade Spacing

For linear cascades, SP is the dimensionless cascade spacing nondimensionalized by the chord of one of the cascades. The chord used to nondimensionalize SP must also be used to nondimensionalize the chords of all the cascades in order to keep the relative size and spacing correct. From this point on, the chord of the body used to nondimensionalize SP will be referred to as the reference chord c_{ref} .

For circular cascades, SP is the angular spacing in degrees.

Only one of the parameters CL, ALPHA, FALPHA, DALPHA needs to be specified, depending on which one is to be used in the combination equations. If not used in the combination equations, any value read is ignored.

CL - Lift coefficient. CL can only be input for linear cascades.

ALPHA - Average flow angle α .

FALPHA - Inlet flow angle α_I .

DALPHA - Change in flow angle through the cascades, $\Delta\alpha$.

SUBKS - A nonzero integer if the unmodified coordinates of all the bodies of the previous case are to be used for the present case. The chords, stagger angles, position of the cascades, etc. may change. For a subcase, the program reads the coordinate data from files generated in the previous run, but then proceeds just as if the coordinate data had been read from cards.

If FLG08 is nonzero (i.e. the program is going directly to the combination equations), the cascade parameter card completes the required input describing the cascades. The next input will be the off-body point data, if the flow at off-body points is to be determined for this case.

Similar Cascade Card: NCSG FORMAT (I5)

NCSG - The number of cascades with geometry similar to cascade 1,
including cascade 1.

This card is only required if several cascades have the same geometry and the coordinates of the similar cascades are to be generated internally subsequent to the input of the coordinates for the first cascade (i.e. this card is not required unless FLG11 is nonzero).

Cascade Geometry Card: CHORD, CCANG, XP, YP, RPADDX, ADDY, NP, NLF,
NLE FORMAT (6F10.0, 3I5)

CHORD - Dimensionless chord of the body. CHORD should be nondimensionalized by the chord of one of the cascades c_{ref} . For a single cascade analysis, CHORD = 1.

CCANG - Stagger angle ξ of the airfoils in the cascade, as illustrated in Figures A1 and A2.

XP, YP - Local coordinates of a point in or on the body which is used to locate the body in the global coordinate system. The body is also rotated around the point (XP, YP) to obtain the desired stagger angle. In addition, the force and moment coefficients are calculated at this point. XP and YP must be nondimensionalized by the chord of the body they are for.

RPADDX - For circular cascades, RPADDX is the radius in the global coordinate system at which the point (XP, YP) is located, as illustrated in Figure A2.

For linear cascades, RPADDX is the x location in the global coordinate system at which the point (XP, YP) is located, as illustrated in Figure A1.

RPADDX must be nondimensionalized by c_{ref} .

ADDY - For circular cascades, ADDY is the angular location in the global coordinate system of the point (XP,YP) for one of the bodies in the cascade. ADDY is illustrated in Figure A2 and must be expressed in degrees.

For linear cascades, ADDY is the y location in the global coordinate system of the point (XP,YP) for one of the bodies in the cascade.

ADDY is illustrated in Figure A1 and must be nondimensionalized by c_{ref} .

NP - Number of points which will be used to represent the body.

NLF - A nonzero integer if the cascade is nonlifting. If lifting and nonlifting bodies are both present in a multiple cascade analysis, the data for the lifting bodies must be read in first.

NLE - For circular cascades with the radial flow outward, NLE is the number of the point which will be the true trailing edge of the airfoil. NLE is not required for the other types of cascades, because the location of the trailing edge is known from the starting point of the data.

Body Coordinate Cards: X, Y

FORMAT (2F10.0)

X, Y - Coordinates of points on the body in the local coordinate system $x'-y'$ which describe the shape of the body. x and y must be nondimensionalized by the chord of the body. There are NP body coordinate cards and the data must start at the points labeled in Figures A1 and A2. For linear cascades and circular cascades with the radial flow inward, the starting point is the trailing edge of the airfoil. The first point must be repeated as the last point.

Off-Body Point Data

Number of Sets Card: NSETS FORMAT (I5)

Set Initial Data Card: NCR, NPS, XR, YT1, DELYT FORMAT (2I5, 3F10.0)

- Zero if the off-body points of this set are to be generated internally.

NPS - Number of points in the set

If NCR is a nonzero integer, XR, YT1 and DELYT do not need to be specified.

XR - For linear cascades, XR is the x coordinate of the off-body points to be generated in this set.

For circular cascades, XR is the r coordinate of the off-body points to be generated in this set.

XR must be nondimensionalized by c_{ref} .

YT1 - For linear cascades, YT1 is the y coordinate of the first point in the set. YT1 must be nondimensionalized by c_{ref} .

For circular cascades, YT1 is the θ coordinate of the first point in the set, expressed in degrees.

DELYT - For linear cascades, DELYT is the y increment by which the remainder of the points in the set will be generated. DELYT must be nondimensionalized by c_{ref} .

For circular cascades, DELYT is the θ increment by which the remainder of the points in the set will be generated. DELYT must be expressed in degrees.

Off-Body Coordinate Cards: XOB, YOB FORMAT (2F10.0)

If the off-body points are to be read in for this set (NCR is nonzero), NPS off-body point coordinate cards are required.

XOB, YOB - For linear cascades, XOB, YOB are the x, y coordinates of the off-body points nondimensionalized by c_{ref} .

For circular cascades, XOB, YOB are the r, θ coordinates, respectively, of the off-body points. XOB must be nondimensionalized by c_{ref} and YOB must be expressed in degrees.

26 June 1981

AMY:cag

The set initial data card and the off-body point coordinate cards when appropriate are required for each set.

The off-body point data completes the description of the various types of input for the program. Data for the next case can follow the present case starting with a new heading card. As previously stated, the program can be terminated by supplying a blank heading card and making FLG12 a nonzero integer on the control card.

APPENDIX A.2

Sample Program Input:

Analytic Cascade Profile C--Multiple Rectangular Cascades

The following sample input data represent the linear cascade and off-body points shown in the Z-plane of Figure 7. The cascade is modeled as two cascades to illustrate a multiple cascade analysis. The coordinates for the second cascade are generated internally (FLG11 = nonzero integer and NCSG = 2), thus, only one set of coordinate data points are given. Relative to the airfoils, the off-body points are the same as those shown in Figure 7 and Table 2. However, since the leading edge of the reference airfoil was positioned at the origin of the global system, the off-body points reflect this position and are translated from those shown in the figure and table.

Data are given for four consecutive runs which demonstrate the programs various input options. The first run has C_L as the given cascade operating parameter. (Note: $C_L = 2(1.75) = 3.5$, since the cascade is modeled as two similar cascades.) Following the data for the first run, the data for the second run directs the program to go directly to the combination equations using $\Delta\alpha$ as the cascade operating parameter. The third run is a subcase of the previous run (coordinate data are read from a file) and α_I is the given cascade parameter. The final run again directs the program to go directly to the combination equations, this time with α as the cascade operating parameter. The data for these four run options will enable the user to verify that the program is operating properly. The output for the first run only is given in Section A.3. The output for the subsequent runs should be similar.

[illegible]

26 June 1981
AMY:cag

-0.00223	0.01827
-0.00060	0.02667
0.00237	0.03576
0.00669	0.04520
0.01231	0.05467
0.01906	0.06387
0.02676	0.07260
0.03520	0.08069
0.05342	0.09477
0.07236	0.10607
0.09111	0.11499
0.11791	0.12491
0.14262	0.13189
0.16520	0.13690
0.19863	0.14250
0.25330	0.14832
0.29687	0.15099
0.36487	0.15297
0.41915	0.15334
0.46715	0.15303
0.51338	0.15227
0.56164	0.15097
0.61651	0.14872
0.68571	0.14421
0.73031	0.13980
0.75661	0.13640
0.78649	0.13160
0.79965	0.12910
0.81359	0.12614
0.82839	0.12262
0.84409	0.11838
0.86074	0.11324
0.87833	0.10693
0.89676	0.09917
0.90622	0.09463
0.91578	0.08960
0.92250	0.08576
0.93490	0.07791
0.94425	0.07121
0.95328	0.06396
0.96185	0.05622
0.96978	0.04810
0.97692	0.03978
0.98315	0.03149
0.98838	0.02352
0.99259	0.01618
0.99580	0.00982
0.99807	0.00476
0.99947	0.00135
1.00000	0.0
1	
1 15	
-0.25000	0.0
-0.25000	-0.39837

26 June 1981
AMY:cag

-0.25000 -0.79675
0.0 -0.19919
0.0 -0.39837
0.0 -0.59756
0.49235 -0.09729
0.49235 -0.27957
0.49235 -0.46185
1.0 -0.19919
1.0 -0.39837
1.0 -0.59756
1.25 0.0
1.25 -0.39837
1.25 -0.79675

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

2	1	0	1	0	0	0	1	0	0	1	0
1.59350	0.0		0.0		0.0		57.53992		0		0

1
1 15
-0.25000 0.0
-0.25000 -0.39837
-0.25000 -0.79675
0.0 -0.19919
0.0 -0.39837
0.0 -0.59756
0.49235 -0.09729
0.49235 -0.27957
0.49235 -0.46185
1.0 -0.19919
1.0 -0.39837
1.0 -0.59756
1.25 0.0
1.25 -0.39837
1.25 -0.79675

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

2	1	0	0	1	0	0	0	0	0	1	0
1.59350	0.0		0.0		28.108		0.0			1	

2											
1.0	0.0		0.0		0.0		0.0		0.0		99 0

1
1 15
-0.25000 0.0
-0.25000 -0.39837
-0.25000 -0.79675
0.0 -0.19919
0.0 -0.39837
0.0 -0.59756
0.49235 -0.09729
0.49235 -0.27957
0.49235 -0.46185
1.0 -0.19919
1.0 -0.39837
1.0 -0.59756
1.25 0.0
1.25 -0.39837

26 June 1981
AMY:cag

```

1.25      -0.79675
ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES
  2      1      1      0      0      0      0      1      0      0      1      0
1.59350   0.0      -0.86154   0.0      0.0      0.0      0.0      0.0
  1
  1      15
-0.25000   0.0
-0.25000  -0.39837
-0.25000  -0.79675
  0.0      -0.19919
  0.0      -0.39837
  0.0      -0.59756
  0.49235  -0.09729
  0.49235  -0.27957
  0.49235  -0.46185
  1.0      -0.19919
  1.0      -0.39837
  1.0      -0.59756
  1.25      0.0
  1.25     -0.39837
  1.25     -0.79675

```

/*

1

APPENDIX A.3

Sample Program Output Using Data in Appendix A.2.

Only the Output Which Uses C_L in the Combination
Equations is included.

26 June 1981
AMY:cag

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

INPUT PARAMETERS

NUMBER CP BODIES= 2

FIG02 = 1 THE FLOW IS TO BE CALCULATED AT POINTS OFF THE BODIES

FIG03 = 0

FIG04 = 0

FIG05 = 0

FIG06 = 1

CL IS USED IN THE COMBINATION EQUATIONS

FIG07 = 0 THE MATRIX OF INFLUENCE COEFFICIENTS WILL NOT BE PRINTED
FIG08 = 0 THE PROGRAM WILL NOT GO DIRECTLY TO THE COMBINATION SOLUTION
FIG09 = 0 RECTANGULAR CASCADES ARE BEING ANALYZED
FIG10 DOES NOT APPLY TO RECTANGULAR CASCADES

FIG11 = 1 THE FIRST CASCADE GEOMETRY IS REPEATED

CASCADE SPACING = 1.59350

CL = 3.50000

SUBKS = 0 THE FOLLOWING RUN IS NOT A SUBCASE

NUMBER OF CASCADES WITH GEOMETRY OF CASCADE 1 = 2
FOR THE SIMILAR CASCADES

CHORD = 1.00000

CCANG = 0.0

XP = 0.0

YP = 0.0

FEADDX = 0.0

ALDY(1) = 0.0

NF = 99

NIF = 0

NIE = 49

OFF-BODY POINT DATA

THERE IS ONE SET OF OFF-BODY POINTS

FOR SET NUMBER 1, NCR IS NONZERO -- THE OFF-BODY POINTS ARE READ IN

THE EQUATIONS ARE REDUCED IN BLOCKS IN SUBROUTINE PART3

26 June 1981
AMY:cagAPPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02376 CFY = 1.77049 CH = 0.89092

	X	Y	V/VREF	CP
1	1.00000000	0.0	-0.70384020	0.50460899
2	0.99944973	0.00049000	-0.73070890	0.46606451
3	0.99390000	0.00223000	-0.72949964	0.46783030
4	0.9745969	0.00346000	-0.71876711	0.48337388
5	0.93002002	0.00533500	-0.70687461	0.50032830
6	0.9151999	0.00719000	-0.69540650	0.51640981
7	0.9152491	0.00949500	-0.68672115	0.52841407
8	0.9153002	0.01180000	-0.68039507	0.53706259
9	0.9384967	0.01440500	-0.67643332	0.54243797
10	0.97816939	0.01701000	-0.67494488	0.54444945
11	0.97388458	0.01978000	-0.67554516	0.54363877
12	0.96960002	0.02255000	-0.67798328	0.54033870
13	0.9647697	0.02535500	-0.68224096	0.53454727
14	0.95998001	0.02816000	-0.68704975	0.52796268
15	0.95475483	0.03091000	-0.69281507	0.52000731
16	0.94953001	0.03366000	-0.69959927	0.51056087
17	0.94397574	0.03627998	-0.71732759	0.48544115
18	0.93843001	0.03890000	-0.72756767	0.47064531
19	0.93265486	0.04134500	-0.74192286	0.44955051
20	0.92688000	0.04379000		
21	0.92094994	0.04604000		
22	0.91501599	0.04829000		
23	0.9091470	0.05032998		
24	0.90301001	0.05237000		
25	0.89697981	0.05421498		
26	0.89095002	0.05571500		
27	0.88493490	0.05737000		
28	0.87892002	0.05937000		
29	0.87295961	0.06084439		
30	0.86699598	0.06232000		
31	0.86110973	0.06363499		
32	0.85522002	0.06494999		
33	0.84937196	0.06671696		
34	0.84322002	0.06939000		
35	0.83714458	0.07114500		
36	0.83106998	0.07290000		
37	0.82494391	0.07429493		

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CPX = 0.02376 CFX = 1.77049 CM = 0.89092

	X	Y	V/VREF	CP
20	0.78381001	0.07568997	-0.75935006	0.42338753
21	0.77366495	0.07727998	-0.77765006	0.39526039
22	0.75852001	0.07887000	-0.77938056	0.39256597
23	0.73518991	0.08058500	-0.80703884	0.34868836
24	0.71186000	0.08230001	-0.82423431	0.32063782
25	0.67352460	0.08383501	-0.83024865	0.31068718
26	0.62319003	0.08537000	-0.82677096	0.31644982
27	0.59220982	0.08563000	-0.83711070	0.29924572
28	0.5523002	0.08589000	-0.83492833	0.30289471
29	0.52016497	0.08538997	-0.82616585	0.31744999
30	0.48509997	0.08489001	-0.81994599	0.32768857
31	0.4502496	0.08365500	-0.81643343	0.33343649
32	0.41495001	0.08241999	-0.81723201	0.33213186
33	0.37300997	0.08011997	-0.82651597	0.31687140
34	0.34107000	0.07782000	-0.83016306	0.31082934
35	0.31711501	0.07557499	-0.85084230	0.27606744
36	0.29316002	0.07332999	-0.88232833	0.22149676
37	0.26347502	0.07181996	-0.90270585	0.18512219
38	0.23477602	0.07031000	-0.93482667	0.12609911
39	0.20545497	0.06844437		
40	0.17429999	0.06658000		
41	0.14274000	0.06425995		
42	0.11358001	0.06194000		
43	0.08337999	0.05901998		
44	0.05458003	0.05610000		
45	0.02449998	0.05242500		
46	0.00000000	0.04875000		
47	0.11306500	0.03957000		
48	0.10271001	0.03596500		
49	0.09254497	0.03236000		
50	0.08238000	0.02837000		
51	0.07268000	0.02438000		
52	0.06293000	0.02015500		
53	0.05839500	0.01553000		
54	0.05381000	0.01181500		
55	0.04944996	0.00966500		

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02376 CFY = 1.77049 CM = 0.89092

	X	Y	V/VREF	CP
39	0.04509000	0.00763000	-0.97040975	0.05830497
40	0.04100499	0.00573000	-1.01327229	-0.02672005
41	0.03692000	0.00383000	-1.06231403	-0.12851048
42	0.03311500	0.00215000	-1.11350918	-0.23990250
43	0.02935000	0.00047000	-1.15027332	-0.32312870
44	0.02590500	-0.00090500	-1.11928177	-0.25279140
45	0.02246000	-0.00228000	-0.88864422	0.21031147
46	0.01937500	-0.00324500	-0.40194774	0.83843803
47	0.01629000	-0.00421000	0.11178565	0.98750401
48	0.01360500	-0.00467500	0.51474506	0.73503757
49	0.01092000	-0.00514000	0.82048509	0.32480291
50	0.00865500	-0.00500000	1.06626129	-0.13691235
51	0.00633000	-0.00486000	1.26808453	-0.60803795
52	0.00456500	-0.00402500	1.43575001	-1.06137753
53	0.00274000	-0.00319000	1.57192612	-1.47095108
54	0.00137000	-0.00159500	1.67788029	-1.81528187
55	0.00090500	0.0	1.75859642	-2.09266090
56	0.00047000	0.00233500	1.81820393	-2.30586529
57	0.00021500	0.00471000	1.83413792	-2.36406136

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR CASE 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02376 CFY = 1.77649 CM = 0.89092

	X	Y	V/VREF	CP
58	0.05342000	0.09477001	1.86263752	-2.46941953
59	0.06235993	0.10042000	1.85461044	-2.43957901
60	0.07235998	0.10606998	1.80421638	-2.25519657
61	0.08173496	0.11052996	1.75702477	-2.08713531
62	0.09113599	0.11495000	1.70895290	-1.92051983
63	0.10451001	0.11995000	1.64284039	-1.69892406
64	0.11791003	0.12491000	1.55518913	-1.41861248
65	0.13026500	0.12839997	1.48511028	-1.20555210
66	0.14262003	0.13189000	1.43773937	-1.06709385
67	0.15490993	0.13439500	1.39103413	-0.93497562
68	0.16520002	0.13690001	1.36934853	-0.87511444
69	0.17491499	0.13970000	1.36294079	-0.85760689
70	0.18362598	0.14249998	1.36893272	-0.87397671
71	0.1906496	0.14541000	1.38963223	-0.93107700
72	0.20330001	0.14832002	1.43764496	-1.06682301
73	0.21504497	0.14965498	1.47401714	-1.17272568
74	0.22686999	0.15099001	1.51411057	-1.29253006
75	0.23866997	0.15197998	1.55624104	-1.42188549
76	0.2504497	0.15302998	1.59549236	-1.54559517
77	0.2622998	0.15415997		
78	0.2741499	0.15531597		
79	0.2860099	0.15648495		
80	0.2978699	0.15765000		
81	0.3097399	0.15881997		
82	0.3216099	0.15999001		
83	0.3334799	0.16116997		
84	0.3453499	0.16234000		
85	0.3572199	0.16351001		
86	0.3690899	0.16468000		
87	0.3809599	0.16585000		
88	0.3928299	0.16702003		
89	0.4046999	0.16819000		
90	0.4165699	0.16936000		
91	0.4284399	0.17053000		
92	0.4403099	0.17170000		
93	0.4521799	0.17287000		
94	0.4640499	0.17404000		
95	0.4759199	0.17521000		
96	0.4877899	0.17638000		
97	0.4996599	0.17755000		
98	0.5115299	0.17872000		
99	0.5233999	0.17989000		
100	0.5352699	0.18106000		

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GALFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CPX = 0.02376 CFY = 1.77049 CM = 0.89092

	X	Y	V/VREF	CP
77	0.73205001	0.12910002		
	0.80661964	0.12761998	1.61687565	-1.61428642
78	0.81351999	0.12614000		
	0.82396961	0.12437999	1.64109993	-1.69320869
79	0.82339000	0.12261999		
	0.83623951	0.12049997	1.66778278	-1.78149891
80	0.84408998	0.11838001		
	0.85241461	0.11580998	1.69513035	-1.87346649
81	0.86074001	0.11324000		
	0.86353497	0.11008501	1.72218513	-1.96592140
82	0.87832999	0.10693002		
	0.88754463	0.10304999	1.74416828	-2.04212284
83	0.89675999	0.09917003		
	0.90148973	0.09689999	1.77716255	-2.15830612
84	0.90522002	0.09463000		
	0.91099977	0.09211498	1.77846622	-2.16294193
85	0.91378001	0.08960003		
	0.91913986	0.08767998	1.78929901	-2.20159054
86	0.92250001	0.08576000		
	0.92869997	0.08183497	1.76652431	-2.12060738
87	0.93389999	0.07791001		
	0.9357472	0.07455999	1.76271534	-2.10716534
88	0.9424999	0.07121003		
	0.94376480	0.06758499	1.73786259	-2.02016544
89	0.95327997	0.06396002		
	0.95756483	0.06009001	1.70203781	-1.89694260
90	0.96184999	0.05622000		
	0.96581459	0.05215999	1.65176296	-1.72832012
91	0.96374003	0.04810000		
	0.97134757	0.04393998	1.58480549	-1.51160812
92	0.9752001	0.03978000		
	0.98103483	0.03563499	1.50127888	-1.25383759
93	0.98315001	0.03149000		
	0.98576498	0.02750500	1.40130997	-0.96360882
94	0.9883001	0.02352000		
	0.99348471	0.01985000	1.28590584	-0.65355301
95	0.99259001	0.01618000		
	0.99413495	0.01300000	1.15816402	-0.34134388

AD-A103 549

PENNSYLVANIA STATE UNIV UNIVERSITY PARK APPLIED RESE--ETC F/G 20/4
A COMPUTER PROGRAM FOR CALCULATING POTENTIAL FLOW SOLUTIONS FOR--ETC(U)
JUN 81 A M YOCUM N00024-79-C-6043

UNCLASSIFIED

ARL/PSU/TN-81-130

NL

2 of 2
60 A
1-8-81

END
DATE
FILMED
10-81
DTIC

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 1

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02376 CFY = 1.77049 CM = 0.89092

	X	Y	V/VREF	CP
96	0.99530002	0.00982000		
	0.99693489	0.00729000	1.01830292	-0.03694057
97	0.99807000	0.00476000		
	0.99876976	0.00305500	0.86849505	0.24571639
98	0.99947000	0.00135000		
	0.99973488	0.00067500	0.70395666	0.50444502
99	1.00000000	0.0		

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02376 CFX = 1.77059 CK = 0.89097

	X	Y	V/VREF	CP
1	1.00000000	0.0	-0.70382303	0.50463319
2	0.99944973	0.00049000	-0.70382303	0.50463319
3	0.99890000	0.00098000	-0.70382303	0.50463319
4	0.99745589	0.00223000	-0.70382303	0.50463319
5	0.99602002	0.00348000	-0.70382303	0.50463319
6	0.99457965	0.00533500	-0.70382303	0.50463319
7	0.99315199	0.00719000	-0.70382303	0.50463319
8	0.99174451	0.00949500	-0.70382303	0.50463319
9	0.99035302	0.01180000	-0.70382303	0.50463319
10	0.98898467	0.01440500	-0.70382303	0.50463319
11	0.98763923	0.01701000	-0.70382303	0.50463319
12	0.98631658	0.01976000	-0.70382303	0.50463319
13	0.98501602	0.02255000	-0.70382303	0.50463319
14	0.98373837	0.02535500	-0.70382303	0.50463319
15	0.98248301	0.02816000	-0.70382303	0.50463319
16	0.98124866	0.03091000	-0.70382303	0.50463319
17	0.98003494	0.03366000	-0.70382303	0.50463319
18	0.97884300	0.03641000	-0.70382303	0.50463319
19	0.97767274	0.03916000	-0.70382303	0.50463319
20	0.97652486	0.04191000	-0.70382303	0.50463319
21	0.97539891	0.04466000	-0.70382303	0.50463319
22	0.97429494	0.04741000	-0.70382303	0.50463319
23	0.97321201	0.05016000	-0.70382303	0.50463319
24	0.97215002	0.05291000	-0.70382303	0.50463319
25	0.97110793	0.05566000	-0.70382303	0.50463319
26	0.97008598	0.05841000	-0.70382303	0.50463319
27	0.96908302	0.06116000	-0.70382303	0.50463319
28	0.96809901	0.06391000	-0.70382303	0.50463319
29	0.96713490	0.06666000	-0.70382303	0.50463319
30	0.96619073	0.06941000	-0.70382303	0.50463319
31	0.96526658	0.07216000	-0.70382303	0.50463319
32	0.96436243	0.07491000	-0.70382303	0.50463319
33	0.96347828	0.07766000	-0.70382303	0.50463319
34	0.96261413	0.08041000	-0.70382303	0.50463319
35	0.96177000	0.08316000	-0.70382303	0.50463319
36	0.96094585	0.08591000	-0.70382303	0.50463319
37	0.96014170	0.08866000	-0.70382303	0.50463319
38	0.95935755	0.09141000	-0.70382303	0.50463319
39	0.95859340	0.09416000	-0.70382303	0.50463319
40	0.95784925	0.09691000	-0.70382303	0.50463319
41	0.95712510	0.09966000	-0.70382303	0.50463319
42	0.95642100	0.10241000	-0.70382303	0.50463319
43	0.95573685	0.10516000	-0.70382303	0.50463319
44	0.95507270	0.10791000	-0.70382303	0.50463319
45	0.95442855	0.11066000	-0.70382303	0.50463319
46	0.95380440	0.11341000	-0.70382303	0.50463319
47	0.95320025	0.11616000	-0.70382303	0.50463319
48	0.95261610	0.11891000	-0.70382303	0.50463319
49	0.95205200	0.12166000	-0.70382303	0.50463319
50	0.95150785	0.12441000	-0.70382303	0.50463319
51	0.95098370	0.12716000	-0.70382303	0.50463319
52	0.95047955	0.12991000	-0.70382303	0.50463319
53	0.94999540	0.13266000	-0.70382303	0.50463319
54	0.94953125	0.13541000	-0.70382303	0.50463319
55	0.94908710	0.13816000	-0.70382303	0.50463319
56	0.94866300	0.14091000	-0.70382303	0.50463319
57	0.94825885	0.14366000	-0.70382303	0.50463319
58	0.94787470	0.14641000	-0.70382303	0.50463319
59	0.94751055	0.14916000	-0.70382303	0.50463319
60	0.94716640	0.15191000	-0.70382303	0.50463319
61	0.94684225	0.15466000	-0.70382303	0.50463319
62	0.94653810	0.15741000	-0.70382303	0.50463319
63	0.94625395	0.16016000	-0.70382303	0.50463319
64	0.94598980	0.16291000	-0.70382303	0.50463319
65	0.94574565	0.16566000	-0.70382303	0.50463319
66	0.94552150	0.16841000	-0.70382303	0.50463319
67	0.94531735	0.17116000	-0.70382303	0.50463319
68	0.94513320	0.17391000	-0.70382303	0.50463319
69	0.94496905	0.17666000	-0.70382303	0.50463319
70	0.94482490	0.17941000	-0.70382303	0.50463319
71	0.94469075	0.18216000	-0.70382303	0.50463319
72	0.94456660	0.18491000	-0.70382303	0.50463319
73	0.94445245	0.18766000	-0.70382303	0.50463319
74	0.94434830	0.19041000	-0.70382303	0.50463319
75	0.94425415	0.19316000	-0.70382303	0.50463319
76	0.94416999	0.19591000	-0.70382303	0.50463319
77	0.94409584	0.19866000	-0.70382303	0.50463319
78	0.94403169	0.20141000	-0.70382303	0.50463319
79	0.94397754	0.20416000	-0.70382303	0.50463319
80	0.94393339	0.20691000	-0.70382303	0.50463319
81	0.94389924	0.20966000	-0.70382303	0.50463319
82	0.94387509	0.21241000	-0.70382303	0.50463319
83	0.94386094	0.21516000	-0.70382303	0.50463319
84	0.94385679	0.21791000	-0.70382303	0.50463319
85	0.94386264	0.22066000	-0.70382303	0.50463319
86	0.94387849	0.22341000	-0.70382303	0.50463319
87	0.94390434	0.22616000	-0.70382303	0.50463319
88	0.94394019	0.22891000	-0.70382303	0.50463319
89	0.94398604	0.23166000	-0.70382303	0.50463319
90	0.94404189	0.23441000	-0.70382303	0.50463319
91	0.94410774	0.23716000	-0.70382303	0.50463319
92	0.94418359	0.23991000	-0.70382303	0.50463319
93	0.94426944	0.24266000	-0.70382303	0.50463319
94	0.94436529	0.24541000	-0.70382303	0.50463319
95	0.94447114	0.24816000	-0.70382303	0.50463319
96	0.94458699	0.25091000	-0.70382303	0.50463319
97	0.94471284	0.25366000	-0.70382303	0.50463319
98	0.94484869	0.25641000	-0.70382303	0.50463319
99	0.94499454	0.25916000	-0.70382303	0.50463319
100	0.94515039	0.26191000	-0.70382303	0.50463319

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BCDY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02370 CFY = 1.77059 CM = 0.89097

	X	Y	V/VREF	CP
20	0.78381001	0.07568997	-0.75934756	0.42339134
	0.77366495	0.07727998		
21	0.75052001	0.07887000	-0.77763969	0.39527655
	0.73518991	0.08058500		
22	0.71186000	0.08230001	-0.77937162	0.39257991
	0.67052460	0.08383501		
23	0.62919003	0.08537000	-0.80703115	0.34870076
	0.59220962	0.08563000		
24	0.55223002	0.08589000	-0.82422429	0.32065433
	0.52016497	0.08538997		
25	0.49309997	0.08489001	-0.83023489	0.31071007
	0.45102436	0.08365500		
26	0.41495001	0.08241999	-0.82675970	0.31646842
	0.37800997	0.08011997		
27	0.34107000	0.07782000	-0.83710212	0.29926008
	0.31711501	0.07557499		
28	0.29316002	0.07332999	-0.83491302	0.30292028
	0.26779002	0.07031000		
29	0.25454497	0.06844437	-0.82614827	0.31747907
	0.24129599	0.06658000		
30	0.22744000	0.06425995	-0.81993353	0.32770902
	0.21358001	0.06194000		
31	0.19907999	0.05901998	-0.81642514	0.33345002
	0.18452003	0.05610000		
32	0.16948998	0.05242500	-0.81722307	0.33214647
	0.15439999	0.04875000		
33	0.13891000	0.04415998	-0.82650405	0.31689107
	0.12342000	0.03957000		
34	0.11306500	0.03596500	-0.83014804	0.31085426
	0.10271001	0.03236000		
35	0.09254497	0.02837000	-0.85083050	0.27608752
	0.08423800	0.02438000		
36	0.07268000	0.02015500	-0.88231504	0.22152019
	0.06293000	0.01593000		
37	0.05331500	0.01381500	-0.90269220	0.18514681
	0.05131000	0.01170000		
38	0.041944998	0.00966500	-0.93480885	0.12613243

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CPX = 0.02370 CPY = 1.77059 CM = 0.89097

	X	Y	V/VREF	CP
39	0.04509000	0.00763000	-0.97039366	0.05833620
40	0.04103499	0.00573000	-0.97039366	0.05833620
41	0.03692000	0.00383000	-1.01325607	-0.02668762
42	0.03313500	0.00215000	-1.06228924	-0.12845802
43	0.02935000	0.00047000	-1.11348057	-0.23983860
44	0.02590500	-0.00090500	-1.15023422	-0.32303810
45	0.02246000	-0.00228000	-1.11924267	-0.25270367
46	0.01937500	-0.00324500	-0.88859361	0.21040142
47	0.01629000	-0.00421000	-0.40189087	0.83348375
48	0.01360500	-0.00467500	0.11182559	0.98749506
49	0.01092000	-0.00514000	0.51478738	0.73499399
50	0.00865500	-0.00500000	0.82050055	0.32677889
51	0.00639000	-0.00486000	1.06628323	-0.13695908
52	0.00410000	-0.00402500	1.26812077	-0.60812950
53	0.00274000	-0.00319000	1.43578815	-1.06148720
54	0.00137000	-0.00159500	1.57197666	-1.47111034
55	0.00000000	0.00000000	1.67792320	-1.81542587
56	0.00000000	0.00000000	1.75863838	-2.09280872
57	0.00000000	0.00000000	1.81824684	-2.30602074
	0.04430997	0.08772999	1.83418369	-2.36422920

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C---MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CPX = 0.02376 CPY = 1.77059 CM = 0.89097

	X	Y	V/VREF	CP
58	0.05342000	0.09477001	1.86267471	-2.46955681
	0.00288993	0.10042000		
59	0.07235998	0.10606998	1.85464478	-2.43970680
	0.08173496	0.11052996		
60	0.04110993	0.11499000	1.80425072	-2.25532055
	0.10451001	0.11995000		
61	0.11791003	0.12491000	1.75706005	-2.08725929
	0.11026500	0.12839997		
62	0.14262003	0.13185000	1.76898056	-1.92061424
	0.15190998	0.13435500		
63	0.16520000	0.13690001	1.64287281	-1.69903088
	0.1191499	0.13970000		
64	0.19862998	0.14249998	1.55521774	-1.41879213
	0.22596496	0.14541000		
65	0.25130001	0.14832002	1.48512936	-1.20560837
	0.27508497	0.14965498		
66	0.29696999	0.15099001	1.43775368	-1.06713486
	0.33086997	0.15197598		
67	0.36487001	0.15297002	1.39104176	-0.93499660
	0.39200997	0.15315497		
68	0.41914999	0.15333998	1.36936092	-0.87514877
	0.44314998	0.15318495		
69	0.46714997	0.15302998	1.36295128	-0.85763550
	0.49264995	0.15265000		
70	0.51137999	0.15227002	1.36894608	-0.87401295
	0.53750992	0.15161997		
71	0.56164002	0.15096998	1.38964558	-0.93111420
	0.58907461	0.14984500		
72	0.61650997	0.14872003	1.43766117	-1.06686878
	0.65110970	0.14646500		
73	0.65571001	0.14420998	1.47403240	-1.17271145
	0.70800972	0.14200497		
74	0.73031002	0.13980001	1.51412487	-1.29257393
	0.74345970	0.13809997		
75	0.75680996	0.13639998	1.55625534	-1.42193031
	0.77154970	0.13400000		
76	0.79306994	0.13160002	1.55548950	-1.54550659
		0.13034999		

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BODY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 20.1791 EXIT FLOW ANGLE = -29.3624

CFX = 0.02370 CFX = 1.77059 CM = 0.89097

	X	Y	V/VREF	CP
77	0.75965001	0.12910002		
	0.80661904	0.12761998	1.61688614	-1.61432076
78	0.81353999	0.12614000		
	0.82398961	0.12437999	1.64111042	-1.69324303
79	0.82839000	0.12261999		
	0.83623981	0.12049997	1.66781425	-1.78160381
80	0.84408998	0.11838001		
	0.85243461	0.11580998	1.69517422	-1.87361526
81	0.86074301	0.11324000		
	0.86953497	0.11008501	1.72222519	-1.96605873
82	0.87832999	0.10693002		
	0.88754463	0.10304999	1.74420547	-2.04225254
83	0.89675959	0.09917003		
	0.90148973	0.09689999	1.77721024	-2.15847588
84	0.90622002	0.09463000		
	0.91099977	0.09211498	1.77850723	-2.16308784
85	0.91573301	0.08960003		
	0.9193986	0.08767998	1.78933525	-2.20172024
86	0.92250001	0.08576000		
	0.92869997	0.08183497	1.76656914	-2.12076569
87	0.93489999	0.07791001		
	0.93957472	0.07455999	1.76275826	-2.10731602
88	0.94424999	0.07121003		
	0.94676480	0.06758499	1.73792171	-2.02037144
89	0.95327597	0.06396002		
	0.95756483	0.06009001	1.70208073	-1.89707851
90	0.96184999	0.05622000		
	0.96581459	0.05215999	1.65181255	-1.72848415
91	0.96978003	0.04810000		
	0.97334957	0.04393998	1.58485699	-1.51177120
92	0.97692001	0.03976000		
	0.98034463	0.03563499	1.50131893	-1.25395775
93	0.98315001	0.03145000		
	0.98576498	0.02750500	1.40135956	-0.96387806
94	0.98833001	0.02352000		
	0.99043471	0.01985000	1.28594971	-0.65366650
95	0.99259001	0.01618000		
	0.99414498	0.01300000	1.15818882	-0.34140110

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR BCDY 2

AVG. FLOW ANGLE = -0.7700 INLET FLOW ANGLE = 28.1791 EXIT FLOW ANGLE = -29.3624

CPX = 0.02376 CRY = 1.77059 CM = 0.89097

	X	Y	V/VREF	CP
96	0.99580302	0.00982000		
	0.99693489	0.00729000	1.01835823	-0.03705311
97	0.99807000	0.00476000		
	0.99876976	0.00305500	0.86853915	0.24563974
98	0.99947600	0.00135000		
	0.99974488	0.00067500	0.70390213	0.50452185
99	1.00000000	0.0		

TOTAL CL FOR ALL THE CASCADES = 3.50007

26 June 1981
AMY:cag

APPLIED RESEARCH LABORATORY
GARFIELD THOMAS WATER TUNNEL

ANALYTIC CASCADE PROFILE C--MULTIPLE RECTANGULAR CASCADES

RESULTS FOR THE OFF BODY POINTS

	X	Y	VX	VY	CP
1	-0.25000	0.0	0.96626	0.55184	-0.23818
2	-0.25000	-0.39837	1.02386	0.51952	-0.31818
3	-0.25000	-0.79675	0.96624	0.55185	-0.23816
4	0.0	-0.19919	0.97556	0.38315	-0.09853
5	0.0	-0.39837	1.11061	0.43456	-0.42229
6	0.0	-0.59756	1.22890	0.63731	-0.91637
7	0.49235	-0.09729	0.55099	0.00803	0.09554
8	0.49235	-0.27957	1.08267	-0.00281	-0.17219
9	0.49235	-0.46185	1.23047	-0.01331	-0.51424
10	1.00000	-0.19919	0.94276	-0.43069	-0.07429
11	1.00000	-0.39837	1.08239	-0.45915	-0.38238
12	1.00000	-0.59756	1.18786	-0.61748	-0.79230
13	1.25000	0.0	0.98062	-0.58214	-0.30050
14	1.25000	-0.39837	1.01896	-0.51452	-0.33480
15	1.25000	-0.79675	0.98061	-0.58214	-0.30049

APPENDIX B

B.1 Data Storage and Equation Solution Procedures

Appendix B is included in this report because the data storage space required by the program is one of the primary concerns when adapting the program to a particular computer system and/or a particular problem. A discussion of the available methods for solving the system of equations is included, because the method used depends on the size of the problem and the amount of core storage available. Although the program is quite general, the purpose of this appendix is to give the program user an understanding of the storage requirements and equation solution techniques so that he can modify the program to meet his needs in the most economical and time efficient manner. For example, if the user has large problems but a computer with a small core; the array dimensions will have to be changed to match the available space and the solution procedure selected which solves the equations with the terms in the equations read from files or auxiliary storage. On the other hand, if relatively small problems are run, the entire matrix representing the system of equations may be stored in the core and reduced in a conventional manner for the solution.

The major portion of the calculations performed in the program are done to solve the system of simultaneous equations which yield the source distributions on the surface of the airfoils. Equation (11) in the main body of this report represents the system of equations in matrix form. To aid in the discussion, Equation (11) is rewritten below:

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ A_{N1} & A_{N2} & & A_{NN} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \cdot \\ \cdot \\ \cdot \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \vec{-V}_{\infty 1} \cdot \vec{n}_1 \\ \vec{-V}_{\infty 2} \cdot \vec{n}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{-V}_{\infty N} \cdot \vec{n}_N \end{bmatrix}$$

or

(B.1)

$$[A] [\sigma] = [O]$$

where $[A]$ = Influence coefficient matrix

$[\sigma]$ = Source distribution matrix (solution matrix)

$[O]$ = Onset flow matrix

As discussed in the main body of the report, several basic solutions are required for each problem. Thus, Equation (B.1) must actually be solved several times. During each step of the solution, the solution matrices for each basic solution are modified; thus, the several required basic solutions are essentially obtained simultaneously.

Before discussing the solution techniques, an important difference should be noted between the A matrix of Equation (B.1) and the matrices of typical finite element and finite difference methods. The sources applied to the surface elements of the airfoils in the Douglas Neumann method have an induced effect on every element. The coefficients in the A matrix in Equation (B.1) represent this influence; and if there are N elements, there are N rows in the matrix each with N terms. Typical finite difference and finite element methods yield matrices with mostly zeros, except near the diagonal. For the finite difference and finite element methods, only

the nonzero terms near the diagonal need to be stored and manipulated for a solution. In comparison, the terms in the A matrix are all nonzero, and therefore N surface elements result in N^2 terms which must be stored and manipulated for the solution. With N^2 terms, it is easy to see that for a large number of surface elements representing the cascades, storage problems and long computer solution times may be encountered.

There are currently three methods available to solve the system of equations in the rewritten version of the Douglas Neumann program. Two of the methods were in the original Douglas Neumann cascade program and a third method was developed by ARL which can handle very large problems and requires less computer time. In the following paragraphs, the three methods will each be explained and their core storage requirements discussed. After the three methods are described, an example illustrating the computation time required by the three methods will be given. The second part of the appendix then specifically discusses the array dimensions in the program and how these dimensions and the equation solution procedure can be changed.

The first method to be discussed for solving the system of simultaneous equations was in the original Douglas Neumann cascade program. It is a matrix reduction technique which requires that the entire A matrix of Equation (B.1) be stored in an array in the computer core. For each basic solution required, there is an onset flow matrix [0] which must also be in the core if all the basic solutions are to be obtained with a single reduction of the A matrix. If there are N elements and N_s basic solutions needed, the total array area required by this reduction technique is $N^2 + N_s N$. With an array area of $N^2 + N_s N$

required, the reduction technique is usually not practical for problems with N greater than two hundred. One hundred elements is usually sufficient to represent one airfoil, thus two cascades can easily be represented by less than two hundred elements and the equations solved using the matrix reduction technique.

The idea behind the typical matrix reduction type of simultaneous equation solution is to operate on the coefficient matrix and the matrix on the right hand side of the equation until the coefficient matrix consists of ones on the diagonal and zeroes everywhere else. When the coefficient matrix is reduced to this form, the matrix on the right hand side of the equation is the solution. Since each row of the two matrices represents an equation, rows can be multiplied by constants and added or subtracted from each other just as equations are manipulated. As the coefficient matrix is reduced, the same operations are performed on the matrix on the right hand side of the equation. However, the values in the matrix on the right hand side do not effect the reduction of the coefficient matrix. For this reason, in the Douglas Neumann program all of the O matrices can be operated on during the reduction of the A matrix, and all the basic solutions are obtained simultaneously.

In a matrix reduction type of solution procedure, it is known to be good practice to have one of the larger values of the row on the diagonal. Since the rows are first divided by the value on the diagonal, a zero cannot be on the diagonal and small values on the diagonal may result in roundoff errors. In the Douglas Neumann program, rather than first rearranging the matrix for a strong diagonal, and equivalent procedure is used where the largest value of the row is used to divide the other values of row and then eliminate other terms in the same column. The result is a one in each column

with the remaining values zero. However, since the largest values are not necessarily on the diagonal, the solution when the reduction is completed is not in the correct order. The second part of the solution technique thus consists of rearranging the solution matrix so the values are in the correct order. One improvement the author of this report feels could be made would be to use a more efficient method of putting the solution in its proper order.

The second method available for solving the set of simultaneous equations was also in the original version of the Douglas Neumann cascade program and is found in one version of Subroutine PART3. A method was required which could handle larger problems where the A matrix was too large to be stored in the core of the computer. In the original Douglas Neumann program an iterative technique was used which only requires one row of the A matrix to be in the computer core at one time. A row of the A matrix is read from auxiliary storage and multiplied by the solution matrix $[\sigma]$ as it currently appears ($[\sigma]$ is initially zero). If the solution matrix is correct, the value resulting from the multiplication will equal the corresponding value in the onset flow matrix. The difference in the two values divided by A_{II} indicates the amount σ_I should be changed for the current iteration. The value of σ_I is then updated and while a particular row of the A matrix is in the core the process is repeated for the other basic solutions. When the calculations are complete for all of the basic solutions, the next row of the A matrix is read from auxiliary storage and the procedure is followed until all the rows have been treated. At the end of a complete iteration the maximum change in any value of σ is compared to a convergence criterion and the iterations continue until the criterion is met. Since different basic solutions may converge with a different number of iterations, the iterations only continue for those basic solutions which have not converged.

The main disadvantage of the iterative approach is that the A matrix must be read from auxiliary storage for every iteration. Since reading files is a much slower process than using values in the computer core, the computation time for the iterative method can be large.

For the purpose of reducing computation time for large problems, a third method for solving the set of simultaneous equations was developed at ARL during the current rewriting of the Douglas Neumann program. The third method employs a simple matrix reduction technique; but for large problems where all the equations cannot fit in the core simultaneously, it combines the use of auxiliary storage and computer core and operates on only a partial block of the equations in the core at a given time. The technique is designed to maximize the use of whatever core space is available and, thus, the dimension of the array used in the reduction can be made whatever size is best for the computer system.

In the third method, given a work space of a certain size, the subroutine first sets aside a portion of the array sufficient to hold one equation. This space is needed for reading the writing data on the files. With the remaining space, the subroutine determines how many equations can fit into the array and reads this number of equations from the file. The block of equations in the core are then partially reduced, resulting in ones at the locations representing the diagonal of the complete matrix and zeros for all the other values in the columns with ones. This partial reduction of the equations in the core is illustrated schematically in Figure B.1 where 4 rows of a small 16 x 16 matrix are shown.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & X & X & X & X & X & X & X & X & X & X & X & X \\ 0 & 1 & 0 & 0 & X & X & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 1 & 0 & X & X & X & X & X & X & X & X & X & X & X & X \\ 0 & 0 & 0 & 1 & X & X & X & X & X & X & X & X & X & X & X & X \end{bmatrix}$$

Figure B.1 Illustration of How a Block of Equations Represented by a Portion of a 16 x 16 Matrix is Partially Reduced in the Core. The X's Represent Nonzero Numbers.

[illegible]

Rows of the matrix
in the file

Figure B.2 Schematic of a 16 x 16 Matrix After it has been Partially Reduced by the First Four Rows of the Matrix.

After each complete step, the subroutine reads in the next group of equations (or rows of the matrix) and repeats the reduction process until all the equations have been in the reduction block in the core and the matrix is completely reduced. The method of reducing the matrix by operating on blocks of equations which are in the core of the computer is expected to result in a substantial savings in computer time over the iterative method, if the number of times the equations must be read from file is less for the reduction method. The number of times the equations must be read from a file is determined by the number of equations which will fit into the allotted space in the core. To save time, the subroutine taken advantage of the fact that it need not operate on or store columns of the matrix which have already been reduced to ones and zeroes. Thus, as the solution proceeds and the number of terms in the equations is decreasing, a larger number of equations will fit into the allotted area, decreasing the required number of steps. Figure B.3 illustrates the second step of the reduction for the example 16 x 16 matrix and also demonstrates how the number of equations which will fit into the core increases as the number of terms decreases.

To better realize the significance of increasing the number of equations in the core as the number of terms decreases, a more realistic example will be given. Consider the case where there are 200 equations each with 200 terms and the allotted array space for solving the equation is 50 x 50 or 2500. Initially 200 spaces are required for reading and writing data on files, thus only 2300 spaces are available for storing equations in the core. With 200 terms per equations, 11 equations can be stored in the core and 11 columns reduced. If the solution technique did not take advantage of the fact that after each step the number of terms has decreased, it would require $200 \div 11 = 18.2$ or 19 steps to reduce the entire matrix. However, with the number of terms decreased each step, more equations fit into the

1 0 0 0	0 0 0 0 0	X X X X X X X	}	Stored in a file
0 1 0 0	0 0 0 0 0	X X X X X X X		
0 0 1 0	0 0 0 0 0	X X X X X X X		
0 0 0 1	0 0 0 0 0	X X X X X X X		
0 0 0 0	1 0 0 0 0	X X X X X X X	}	Segment of the matrix in the core for the second step of the reduction
0 0 0 0	0 1 0 0 0	X X X X X X X		
0 0 0 0	0 0 1 0 0	X X X X X X X		
0 0 0 0	0 0 0 1 0	X X X X X X X		
0 0 0 0	0 0 0 0 1	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X	}	Stored in a file
0 0 0 0	0 0 0 0 0	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X		
0 0 0 0	0 0 0 0 0	X X X X X X X		

columns not
not
stored
after
1st
step

columns not
stored
after
2nd step

Figure B.3 Schematic of the Example 16 x 16 Matrix After the Second Step of the Reduction. The Brackets Show the Block of the Matrix Which was in the Core for the Second Step.

core and only 10 steps are required. Table B.1 depicts the steps required for this example. Ten steps, meaning ten times the matrix must be read from files compared to approximately 20 iterations required for the iterative approach would result in a substantial savings in computer time.

One potential disadvantage of reducing the matrix by blocks of equations in the core is that the technique does not utilize the concept of first arranging the matrix for a strong diagonal. In cases run to date, this has not been found to cause any problems.

Now that the three solution techniques and their core storage requirements have been discussed, a comparison of the computation time for each of the techniques for a sample problem is appropriate. For the analytical airfoil profile in a linear cascade previously discussed in the section on check cases, the program was run with the equations solved using each of the three techniques. The cascade was modeled as two similar cascades to increase the number of points and to check the multiple cascade solution procedure. With two cascades each with 99 points describing the airfoils, there was a total of 196 equations each with 196 terms. Since there were two cascades with circulation, four basic solutions were required. The reduction technique with all the equations in the core utilized an array area 200 x 200. The reduction of the matrix with only a block of equations in the core was performed using an array area 100 x 100. Constructing a table similar to Table B.1 would show that the matrix was reduced in three steps. For the iterative technique, the number of iterations required by each of the basic solutions is shown in Table B.2 along with the computation times required by the three techniques. Results for the iterative technique for two different convergence criteria are shown in the table.

TABLE B.1

Steps in Reducing a 200 x 200 Matrix
in a 50 x 50 Array

Step	Terms per Equation	Array Space Available for Equations	Equations in the Core and Columns Reduced	Total Number of Columns Reduced
1	200	2300	11	11
2	189	2311	12	23
3	177	2323	13	36
4	164	2336	14	50
5	150	2350	15	65
6	135	2365	17	82
7	118	2382	20	102
8	98	2402	24	126
9	74	2426	32	158
10	42	2458	58 Maximum 42 Needed	200

TABLE B.2

A Comparison of the Computation Times Required by the
Three Solution Procedures for a Sample Problem

	Reduction with complete matrix in the core	Reduction by blocks of equations in the core	Iterative Procedure	
			Convergence Criterion = .01 Iterations = 21, 31, 21, 21	Convergence Criterion = .001 Iterations = 26, 41, 27, 26
CPU Time (sec)	232.55	135.44	102.58	123.82
I/O Time (sec)	21.54	31.03	147.51	189.50
Total Time (sec)	254.09	166.47	250.09	313.32

It is seen from Table B.2 that the procedure which reduces the matrix by blocks of equations in the core required the least amount of time of the three techniques. The reduction in blocks procedure required significantly less CPU time than the procedure with the complete matrix in the core for two reasons: (1) the procedure which reduces the matrix by blocks does not even consider reduced columns where as the other technique merely skips zero terms when encountered, (2) The procedure which reduces the complete matrix in the core must reorder the solution when the reduction is completed. It is also seen from Table B.2 that the reduction in blocks technique requires considerably less computer time than the iterative technique because of its savings in I/O time. I/O (i.e. input/output) time reflects the time spent reading disc files. The iterative technique must read the complete set of equations from a file for each iteration; and as shown in the table, for a convergence criteria of .001 the second basic solution required 41 iterations.

To conclude this section of Appendix B, Table B.3 summarizes the array storage requirements and the major advantages and disadvantages of the three solution procedures. With a general understanding of the three procedures and there storage requirements from this section, the following section explains specifically the array dimensions and how they and the solution procedure can be changed.

TABLE B.3
A Comparison of the Three Solution Procedures

Solution Procedure	Array Storage Spaces Required	Major Advantages	Major Disadvantages
Reduction with the complete matrix in the core	N^2	Utilizes strong diagonal concept	1. Limited to small problems because of storage requirements 2. Longer computation time
Reduction by blocks of equations in the core	At least $2N$, larger storage areas will reduce the computation time	1. Shorter computation time 2. Can utilize whatever storage space is available	Does not utilize strong diagonal concept
Iterative Procedure	N	Small array storage requirements	Longer computation time

N = number of equations (and number of terms per equation)

B.2 Changing Array Dimensions and Solution Procedures

It may sometimes be necessary to change the array dimensions to match the program to a particular computer system or a particular problem. Changing solution procedures primarily involves making certain the array dimensions are consistent with the procedure, thus both changing array dimensions and solution procedures will be discussed in the context of explaining the arrays in the program. Several segments of the computer program are presented in figures to aid in the discussion. All the major arrays used in the program are in common statements, utilizing the same storage space for all the subroutines and transferring the needed values from subroutine to subroutine.

Figure B.4 is a listing of the main program of the rewritten version of the Douglas Neumann program which reveals all the major arrays used. The COMMON statements BLK2 and BLK3 contain the arrays which describe the geometry of the airfoils in the cascades. The variable arrays in these statements are defined below:

X, Y - the coordinates input to the program describing the shape
of the airfoils

XMP, YMP - the coordinates of the midpoints of the surface elements
defined by X, Y

Q - complex body coordinates ($X + iY$)

SINA, COSA - sine and cosine of the angle of the surface elements

Since the arrays in the COMMON statements BLK2 and BLK3 contain airfoil coordinate data, it is obvious that they must be dimensioned as large as the number of points used to describe the shape of the airfoils in the cascades. The current version of the program was intended to handle up to 10 different cascades each of which can usually be adequately described with 100 points or less. These numbers (10 x 100) resulted in


```

1      COMMON/BLK1/HEDR(20),THETA
2      COMMON/BLK2/ Q(1000),SINA(1000),COSA(1000)
3      COMMON/BLK3/ X(1000),Y(1000),XMP(1000),YMP(1000)
   C   COMMON/BLK2/ Q(300),SINA(300),COSA(300)
   C   COMMON/BLK3/ X(300),Y(300),XMP(300),YMP(300)
4      COMMON/BLK4/ NP(10),NLE(10),SUMDS(10),CHORD(10),XP(10),YP(10),
   *RPADDX(10),CCANG(10),ADDY(10),NLF(10)
5      COMMON/BLK5/ FLG02,FLG03,FLG04,FLG05,FLG06,FLG07,FLG08,FLG09,
   *FLG10,FLG11,FLG12
6      COMMON/BLK6/ NB,NT,IM,NSDL,RPI,R2PI,SP,CL,ALPHA,FALPHA,DALPHA
7      COMMON/BLK7/ XOB(500),YOB(500),NOBP
8      COMMON/BLK8/ SIG(1000,12)
   C   COMMON/BLK8/ SIG(300,12)
9      COMMON/BLK9/ A(40000)
   C   ARRAY A IN COMMON BLOCK 9 IS USED BY SEVERAL DIFFERENT
   C   SUBROUTINES TO STORE TEMPORARY VARIABLES
10     COMPLEX IM,Q
11     INTEGER FLG02,FLG03,FLG04,FLG05,FLG06,FLG07,FLG08,FLG09,
   *FLG10,FLG11,FLG12,SUBKS
   C
   C   PART1--READS IN DATA, SCALES, ROTATES AND TRANSLATES THE AIRFOILS.
   C   IT ALSO TRANSFORMS RADIAL FLOW CASCADES
   C
12     10 CALL PART1
13     IF(FLG08.NE.0) GO TO 30
   C
   C   PART2--SETS UP THE MATRIX WHICH IS SOLVED FOR THE SOURCE
   C   DISTRIBUTIONS
   C
14     CALL PART2
15     IF(NT.LE.200) GO TO 20
   C
   C
   C   PART3--SOLVES THE MATRIX BY REDUCING THE EQUATIONS IN BLOCKS
   C   IF THE NUMBER OF ELEMENTS IS GREATER THAN 200
   C
16     CALL PART3
17     GO TO 30
   C
   C   PART4--SOLVES THE MATRIX BY REDUCTION IF THE NUMBER OF ELEMENTS
   C   IS LESS THAN OR EQUAL TO 200
   C
18     20 CALL PART4
   C
   C   PART5--SOLVES THE COMBINATION EQUATIONS, CALCULATES THE VELOCITIES
   C   AND PRESSURES AND WRITES THE OUTPUT
   C
19     30 CALL PART5
20     GO TO 10
21     END

```

Figure B.4 A Listing of the Main Program

the dimension of 100 seen in Figure B.4. Any number of points can be used to describe the airfoils of cascades, provided the total does not exceed the dimensions in COMMON statements BLK2 and BLK3. For many problems, however, a large number of points is not required and the dimensions in COMMON statements BLK2 and BLK3 can be reduced. Shown as comment statements, the COMMON statements with the arrays dimensioned 300 are an example of the dimensions used for several sample runs. This example is carried throughout the other COMMON statements and subroutines to illustrate the changes which must be made to keep all the array dimensions consistent.

COMMON statement BLK4 contains various cascade parameters. Since the program was intended to handle up to 10 cascades, the variables in COMMON/BLK4/ are all dimensioned 10. These dimensions could be reduced for problems with fewer than 10 cascades. However, reducing these dimensions would yield an insignificant savings in storage space and it is not recommended they be reduced. If it every is desirable to analyze more than 10 cascades, the dimension of the variables in COMMON/BLK4/ would have to be increased.

The next major COMMON statement is COMMON/BLK7/, which contains the coordinates (XOB, YOB) of the off-body calculation points. The off-body point coordinates are stored in separate arrays from the body points in the rewritten version of the Douglas Neumann program so that the number of off-body calculation points can be changed easily without affecting the number of body points.

The statement COMMON/BLK8/ SIG(1000,12) dimensions the SIG array which stores the distribution of sources on the surfaces of the airfoils. Each surface element has a constant value of source strength for each of the basic solutions required. The first dimension of SIG must therefore be greater than or equal to the number of surface elements, and the second

dimension must be greater than or equal to the number of basic solutions. If there are 10 cascades with circulation to be analyzed, there are $10 + 2 = 12$ basic solutions required. SIG is therefore dimensioned (1000, 12) in Figure B.4 which is consistent with the previous dimensions in BLK2, BLK3 and BLK4. In the comment common statement, SIG is dimension (300, 12), to be consistent with comment COMMON/BLK3/ and COMMON/BLK4/ shown in the figure. The 300 dimension in BLK3 and the 12 dimension is unchanged since the dimensions in BLK4 are unchanged. In short, the SIG dimensions can be changed, but they should be consistent with the dimensions in the other arrays.

The final COMMON storage area is COMMON/BLK9/. The A array is COMMON/BLK9/ is used to solve the set of simultaneous equations. In SUBROUTINE PART5 the same common space is divided among three arrays which are used to calculate the velocities on the airfoils and reorder the data. Since COMMON/BLK9/ is dimensioned several different ways in the various subroutines, care must be taken to make all changes consistent throughout the program.

The program normally contains two methods of solving the set of simultaneous equations and selects one method according to the size of the problem being run. The equations are either solved by reduction of the complete matrix in an array in SUBROUTINE PART4 and SUBROUTINE MIS1, or the equations are solved by reduction of blocks of equations or by iterating in one of the two versions of SUBROUTINE PART3. The program user can interchange the two versions of SUBROUTINE PART3 as he chooses. The dimension of the A array determines the maximum number of equations which can be solved using the straight matrix reduction. In Figure B.4, the A array is dimensioned 40000 which corresponds to a square matrix 200×200 . Thus, with this dimension, up to 200 equations can be solved by reducing the complete matrix in an array. The statement IF(NT. LE. 200) go to 20 shown in Figure B.4 directs the program to SUBROUTINE PART4 if the number of elements (equations)

is less than or equal to 200. The number in the IF statement may be changed to be less than the square root of the A dimension, but it may not exceed the square root of the A dimension.

Figure B.5 shows the beginning segments of subroutines PART4 and MIS1 and reveals the COMMON statements required by these subroutines. It should be noted that in SUBROUTINE PART4 the A array is two-dimensional and has equal dimensions since it represents a square matrix. Because of its form in SUBROUTINE PART4, it is clear that the dimension of the A array in other subroutines must be such that it can be represented as a square two-dimensional array. It is also shown in Figure B.5 that in SUBROUTINE MIS1 both the A array in COMMON/BLK9/ and the B array in COMMON/BLK8/ are one-dimensional while in the calling subroutine (SUBROUTINE PART4) both arrays are two-dimensional. The parameters ND and ND2 in SUBROUTINE MIS1 provide MIS1 with the first dimension of A and B in their two-dimensional form and enable MIS1 to recover values from the one-dimensional arrays which were stored as values in two-dimensional arrays. If the dimensions of A or B are changed, the parameters ND and ND2 must also be changed.

COMMON/BLK9/ also appears in the two versions of SUBROUTINE PART3. In the version which solves the equations by reducing blocks of equations, the A arrays stores the equations being reduced. The beginning segment of this version of SUBROUTINE PART3 is shown in Figure B.6. It is seen from the figure that the A array is one-dimensional. The parameter NA shown in Figure B.6 must be given the value of the A array dimension in order that the subroutine can calculate the number of equations which will fit into the array. Thus, if the dimension of A is changed, NA must also be changed. The same COMMON statements are also used by the version of SUBROUTINE PART3 which iterates for a solution. No parameters

```

596      SUBROUTINE PART4
      C
      C      IF THE NUMBER OF ELEMENTS IS LESS THAN OR EQUAL TO 200, THIS
      C      SUBROUTINE SETS UP THE MATRICIES IN ARRAYS WHICH ARE THEN SOLVED
      C      BY REDUCTION IN SUBROUTINE MIS1
      C
597      COMMON/BLK2/ Q(1000),SINA(1000),COSA(1000)
      C      COMMON/BLK2/ Q(300),SINA(300),COSA(300)
598      COMMON/BLK6/ NB,NT,IM,NSOL,RPI,R2PI,SP,CL,ALPHA,FALPHA,DALPHA
599      COMMON/BLK8/ SIG(1000,12)
      C      COMMON/BLK8/ SIG(300,12)
600      COMMON/BLK9/ A(200,200)
601      COMPLEX IM,Q

623      SUBROUTINE MIS1(N,NSOL,NERR)
      C      THIS SUBROUTINE REDUCES THE A MATRIX FOR EACH VECTOR IN B.
      C      THE FINAL SOLUTIONS ARE IN B.
624      COMMON/BLK8/ B(12000)
      C      COMMON/BLK8/ B(3600)
625      COMMON/BLK9/ A(40000)
      C      CHANGE ND2 WHEN CHANGING THE DIMENSION OF B
      C      ND2 = (B DIMENSION)/12
      C      ND = SQRT(A DIMENSION)
      C
626      NERR=1
627      ND=200
628      ND2=1000
629      M=NSOL

```

Figure B.5 Parts of Subroutines PART4 and MIS1 Showing the Array Dimensions

```

456      SUBROUTINE PART3
      C
      C      IF THE NUMBER OF EQUATIONS IS GREATER THAN 200,
      C      THIS SUBROUTINE SOLVES THE EQUATIONS BY REDUCTION
      C      HAVING ONLY A PARTIAL BLOCK OF EQUATIONS IN THE CORE
      C      AT ONE TIME. THE NUMBER OF EQUATIONS IN THE BLOCK INCREASES
      C      AS THE NUMBER OF TERMS IS REDUCED, THUS DECREASING THE
      C      STEPS REQUIRED.
      C
457      COMMON/BLK2/ Q(1000),SINA(1000),COSA(1000)
      C      COMMON/BLK2/ Q(300),SINA(300),COSA(300)
458      COMMON/BLK6/ NB,NT,IM,NSOL,RPI,R2PI,SP,CL,ALPHA,FALPHA,DALPHA
459      COMMON/BLK8/ SIG(1000,12)
      C      COMMON/BLK8/ SIG(300,12)
460      COMMON/BLK9/ A(40000)
461      COMPLEX IM,Q
462      REWIND 3
463      REWIND 8
464      WRITE(6,601)
465      601 FORMAT('0',10X,'THE EQUATIONS ARE REDUCED IN BLOCKS ',
      *              'IN SUBROUTINE PART3')
466      DO 30 I=1,NT
467      SIG(I,1)=SINA(I)
468      SIG(I,2)=-COSA(I)
469      IF(NSOL.LT.3) GO TO 30
470      DO 10 J=3,NSOL
471      READ(3) SIG(I,J)
472      10 CONTINUE
473      DO 20 J=3,NSOL
474      SIG(I,J)=-SIG(I,J)
475      20 CONTINUE
476      30 CONTINUE
      C
      C      NEQB = NUMBER OF EQUATIONS IN THE BLOCK, NEQB CHANGES
      C      AS THE NUMBER OF TERMS IS REDUCED..
      C      N1NR = NUMBER OF TERMS IN THE EQUATIONS WHICH HAVE NOT
      C      BEEN REDUCED TO 0 OR 1
      C      NTR = NUMBER OF TERMS PER EQUATION REDUCED TO 0 OR 1
      C      NA = DIMENSION OF THE A ARRAY
      C
      C      THE LAST SEGMENT OF THE A ARRAY IS USED FOR TEMPORARY
      C      STORAGE REQUIRED FOR READING AND WRITING ON FILE.
      C      IFILE = CODE FOR DETERMINING WHICH FILE IS READ AND WHICH
      C      IS WRITTEN ON
      C
477      NA=40000
478      IFILE=8
479      NTR=0
480      40 N1NR= NT-NTR
481      NEQB=(NA-N1NR)/N1NR
482      IF(NEQB.GT.NT) NEQB=NT

```

Figure B.6 Beginning at SUBROUTINE PART3 Showing the COMMON Statements and Important Parameters

need to be changed in the iterative version when array dimensions are changed.

When the set of simultaneous equations have been solved for the source distributions on the airfoils, the storage space in COMMON/BLK9/ is free and can be used for other purposes. This area is therefore used in SUBROUTINE PART5 for calculating velocities and rearranging the data. The beginning segment of SUBROUTINE PART5 shown in Figure B.7 reveals that the storage space in COMMON/BLK9/ is divided among three arrays. The space can be divided in any manner among the array, provided each array has sufficient space to perform its task. The B array is used to store the coefficients for the influence of the sources of all the elements on a particular element. The B matrix is very similar to the A matrix of Equation B.1 except the velocity component calculated from the B matrix is tangent to the body surface rather than normal to the surface. Only one row of the complete B matrix needs to be stored in the B array at one time, thus its dimension can be as small as the total number of surface elements. As the velocities at the midpoints of the elements are calculated, the values are stored in the array V. The velocities are calculated for one cascade and then the results are printed. Therefore, the V array need only store the velocities for one cascade at a time and V must be dimensioned at least as large as the largest number of elements representing any airfoil in an analysis. The array TSTOR is used to temporarily store the velocities and element midpoint coordinates for the cases where the data must be reordered at the end of the computations. TSTOR must also be dimensioned at least large as the largest number of elements representing any airfoil. For example, in a multiple cascade analysis with three cascades represented by 80, 90, and 100 elements, V and TSTOR must both be dimensioned at least 100. For this example B must be dimensioned at least as large as the total

```

637      SUBROUTINE PART5
      C
      C      THIS SUBROUTINE SOLVES THE COMBINATION EQUATIONS AND
      C      CALCULATES THE VELOCITIES AND PRESSURES.
      C
688      COMMON/BLK1/HEDR(20),THETA
689      COMMON/BLK2/ Q(1000),SINA(1000),COSA(1000)
690      COMMON/BLK3/ X(1000),Y(1000),XMP(1000),YMP(1000)
      C      COMMON/BLK2/ Q(300),SINA(300),COSA(300)
      C      COMMON/BLK3/ X(300),Y(300),XMP(300),YMP(300)
691      COMMON/BLK4/ NP(10),NLE(10),SUMDS(10),CHORD(10),XP(10),YP(10),
      *RPADDX(10),CCANG(10),ADDY(10),NLF(10)
692      COMMON/BLK5/ FLG02,FLG03,FLG04,FLG05,FLG06,FLG07,FLG08,FLG09,
      *FLG10,FLG11,FLG12
693      COMMON/BLK6/ NB,NT,IM,NSQL,RPI,R2PI,SP,CL,ALPHA,FALPHA,DALPHA
694      COMMON/BLK7/ X0B(500),Y0B(500),N0BP
695      COMMON/BLK8/ SIG(1000,12)
      C      COMMON/BLK8/ SIG(300,12)
696      COMMON/BLK9/ B(16000),V(12000),TSTOR(12000)
697      COMPLEX IM,Q,CSINH,Z0B,TF,W
698      INTEGER FLG02,FLG03,FLG04,FLG05,FLG06,FLG07,FLG08,FLG09,
      *FLG10,FLG11,FLG12,SURKS
699      DIMENSION DVBAS(10,12),DV(11,11),DVS(11,11),GAM(11)
700      DIMENSION COMBS(12)

```

Figure B.7 Beginning of SUBROUTINE PART5 Showing the Array Dimensions

number of elements or $80 + 90 + 100 = 270$. Under usual circumstances COMMON/BLK9/ will contain much more space than actually required by B, V and TSTOR.

The major arrays in the program have now been discussed and, if it is necessary, the program user should be able to change the dimensions of the arrays. The user can also change the criterion which directs the program to one of the two internal solution procedures or interchange the two versions of SUBROUTINE PART3, since the primary concern when making such changes is providing adequate and consistent array storage. When changing the dimensions of arrays which are used for solving the set of simultaneous equations, the user must remember to change the appropriate parameters in the subroutines as discussed in this appendix.

APPENDIX C

Coordinate Data Input Order and Reordering

The coordinate data describing the shape of the bodies in a cascade are obviously important input to the Douglas Neumann program. Of great importance, although for reasons not as obvious, are the starting point and order of the data. The order of the data is important because the sine and cosine of the angle of the surface elements are calculated from the coordinate data. If the coordinate data are not in the correct order, the sign of the angle according to the chosen convention will be incorrect, and the boundary condition will not be properly satisfied on the body surface. For linear cascades and circular cascades with the flow radially inward, the starting point is also important because it tells the location of the trailing edge. The location of the trailing edge must be known for the program to satisfy the Kutta condition.

Concerning the starting point and order of the body coordinate data, there are three types of cascades which must be given separate consideration. The three types are the linear cascade, the circular cascade with the radial flow inward, and the circular cascade with the radial flow outward. To avoid confusion, it is desirable to keep the starting point and order of the data consistent for all three types of cascades. The order of the data was chosen to match the original Douglas Neumann program for a linear cascade. For the circular cascades, the program must reorder the data so that after the transformation, the starting point and order correspond to the convention for the linear cascade. In this Appendix, the coordinate data input for each type of cascade will be described and the reordering of the data performed by the program will be discussed.

For all three types of cascades, the coordinate data describing the shape of a body must be input to the program in terms of a local coordinate system $x'-y'$. An example of an airfoil in its local $x'-y'$ coordinate system is shown in Figure C1. This airfoil will be used for each type of cascade to illustrate how the local coordinate system for the airfoil is related to the cascade coordinate system. As illustrated in Figure C1, the coordinate data must start at the trailing edge and progress clockwise around the airfoil. (Note: For circular cascades, the orientation of the $x'-y'$ coordinate system with respect to the cascade coordinate system is the same regardless of whether the flow is radially inward or outward. Thus, the true leading edge and trailing edge may be reversed from the orientation shown in Figure C1. Subsequent figures will clarify this point.)

Figure C2 shows how the airfoil and its local coordinate system relate to the cascade coordinate system $x-y$ for the linear type cascade. Although the local coordinate system of the airfoil can be rotated and translated within the global system, the cascade must parallel the y axis and the leading edge of the airfoil must be toward the $-x$ direction. The component of the flow perpendicular to the cascade is from left to right as shown in Figure C2. Since the cascade solution technique is the same for all three types of cascades, the cascade shown in Figure C2 also represents the orientation of the cascade, the direction of flow, and the order and starting point of the data for the radial flow cascades after they are transformed.

The orientation of the local coordinate system for an airfoil in a circular cascade with the radial flow direction inward is shown in Figure C3-a. Again the airfoils can be rotated and located at any $R-\theta$ position, but the origin of the $x'-y'$ system must be facing $R = \infty$. After the cascade is transformed to a linear cascade, the orientation of the airfoils, the

26 June 1981
AMY:cag

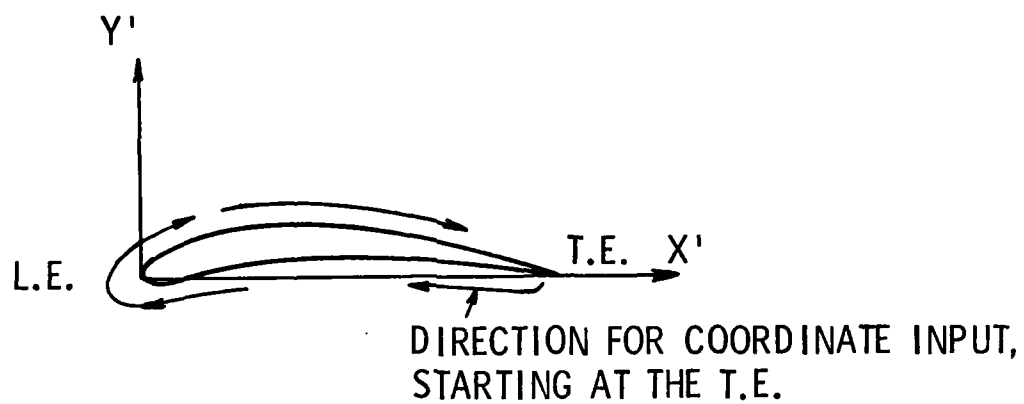


Figure C.1. An Airfoil in its Local Coordinate System $x'-y'$, With the Order for the Data Input Indicated by the Arrows

26 June 1981
AMY:cag

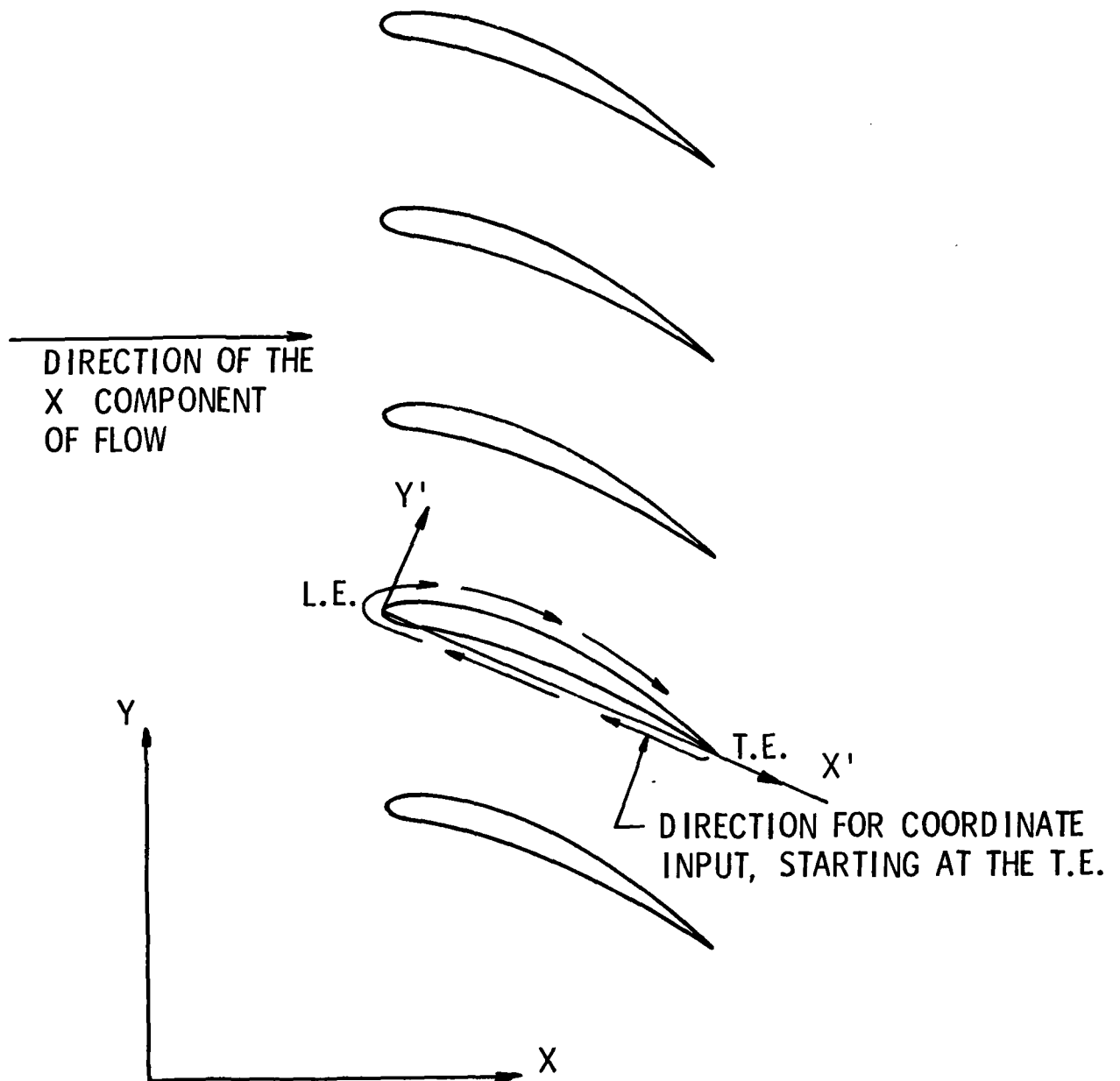


Figure C.2. A Linear Cascade Showing the Orientation of Airfoil Coordinate System With Respect to the Cascade Coordinate System

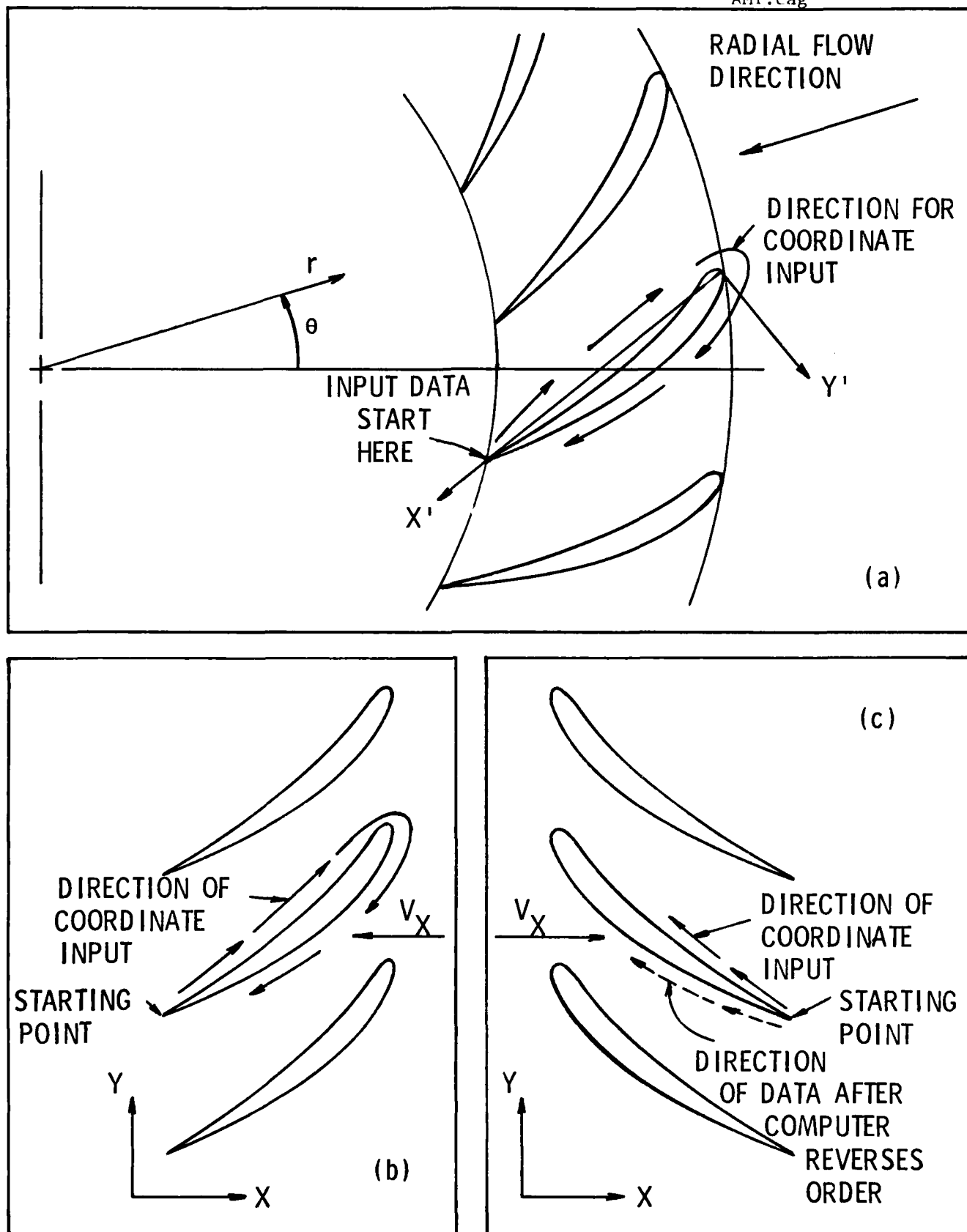


Figure C.3. a. Schematic of a Circular Cascade Showing The Orientation of the Airfoil Coordinate System in the Global System. b. A Schematic Showing the Cascade Orientation and Coordinate Input Direction After the Transformation. c. Final Cascade Configuration and Order of Coordinate Data

order of the coordinate data and the V_x flow direction are shown schematically in Figure C3-b. Figure C3-b reveals that the V_x direction does not correspond to the $-x$ to $+x$ direction required by the cascade solution procedure. For this reason, an additional transformation step is performed to change the sign of the x coordinates of the transformed cascade, resulting in the cascade shown in Figure C3-c. After this step, Figure C3-c shows that the cascade orientation and V_x flow direction are correct, but it is seen that the coordinate data as read in do not correspond to the order chosen for the cascade solution procedure. Rather than having different input orders for different types of cascades, the computer program reverses the order of the coordinate data and the cascade solution proceeds. Following the flow field solution, the order of the data is again reversed so the output order matches the original input.

Figure C4-a shows the orientation of the $x'-y'$ system for an airfoil in the third type of cascade, a circular cascade with the radial flow outward. The $x'-y'$ orientation, order of the coordinate data, and the starting point are the same as for a circular cascade with the radial flow inward. Identical coordinate input for the two types of circular cascades was chosen for the computer program so that wicket gates and stay vanes of hydraulic pump-turbines could be analyzed in both modes with the same coordinate data. A code in the input tells the program the direction of the flow, and for the circular cascade with the radial flow outward the location of the trailing edge is specified since in this case the data do not start there.

When the cascade in Figure C4-a is transformed, the resulting linear cascade is shown schematically in Figure C4-b. It is seen in this figure that the orientation of the airfoils and the direction of V_x are correct

for the cascade analysis. However, the data start at the point which would be the trailing edge if the flow was radially inward, and the starting point must be moved to the true trailing edge for the radial flow outward condition. Again the computer program reorders the data so that the input can be consistent for each type of cascade. For the circular cascade with the radial flow outward, the index of the coordinate pair which give the location of the trailing edge is input to the program, and this point is made the starting and finishing point. As in the case when the radial flow is inward, the output is put back into the original order after the flow field calculations are complete.

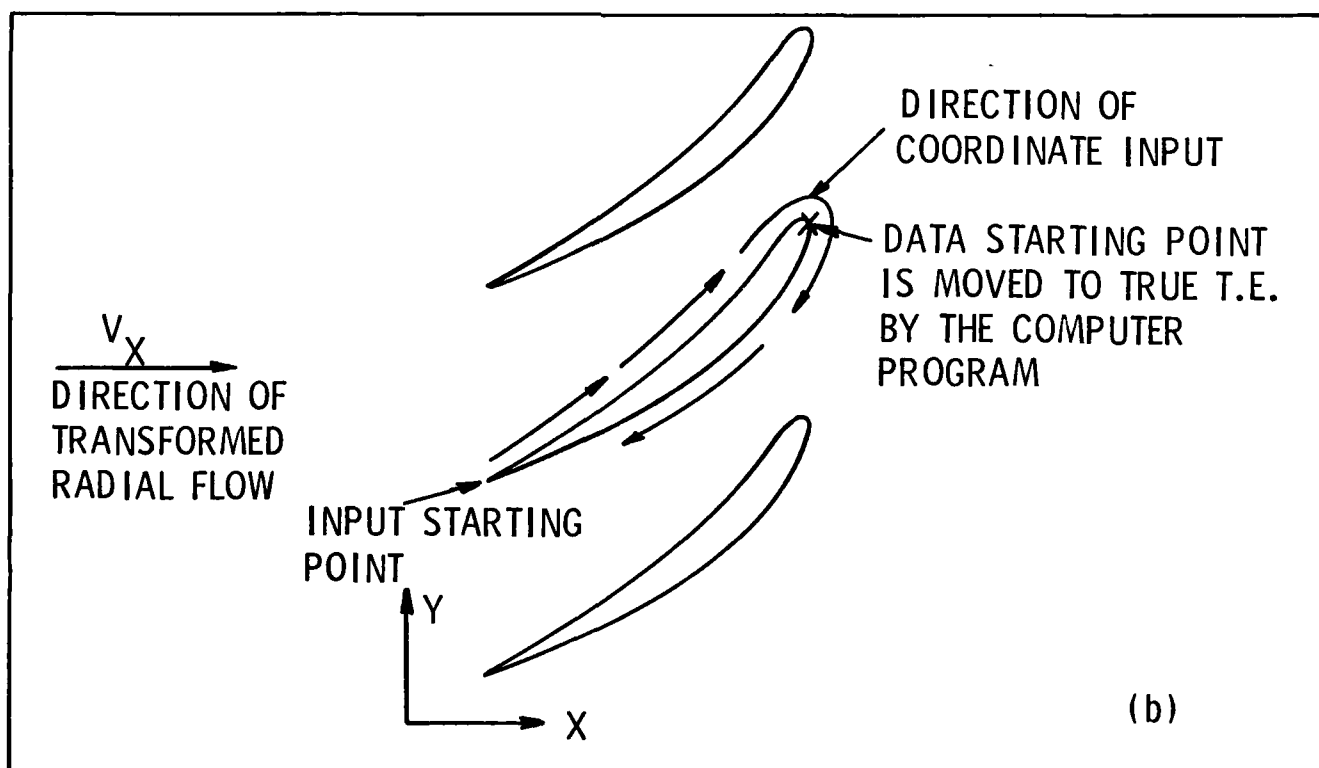
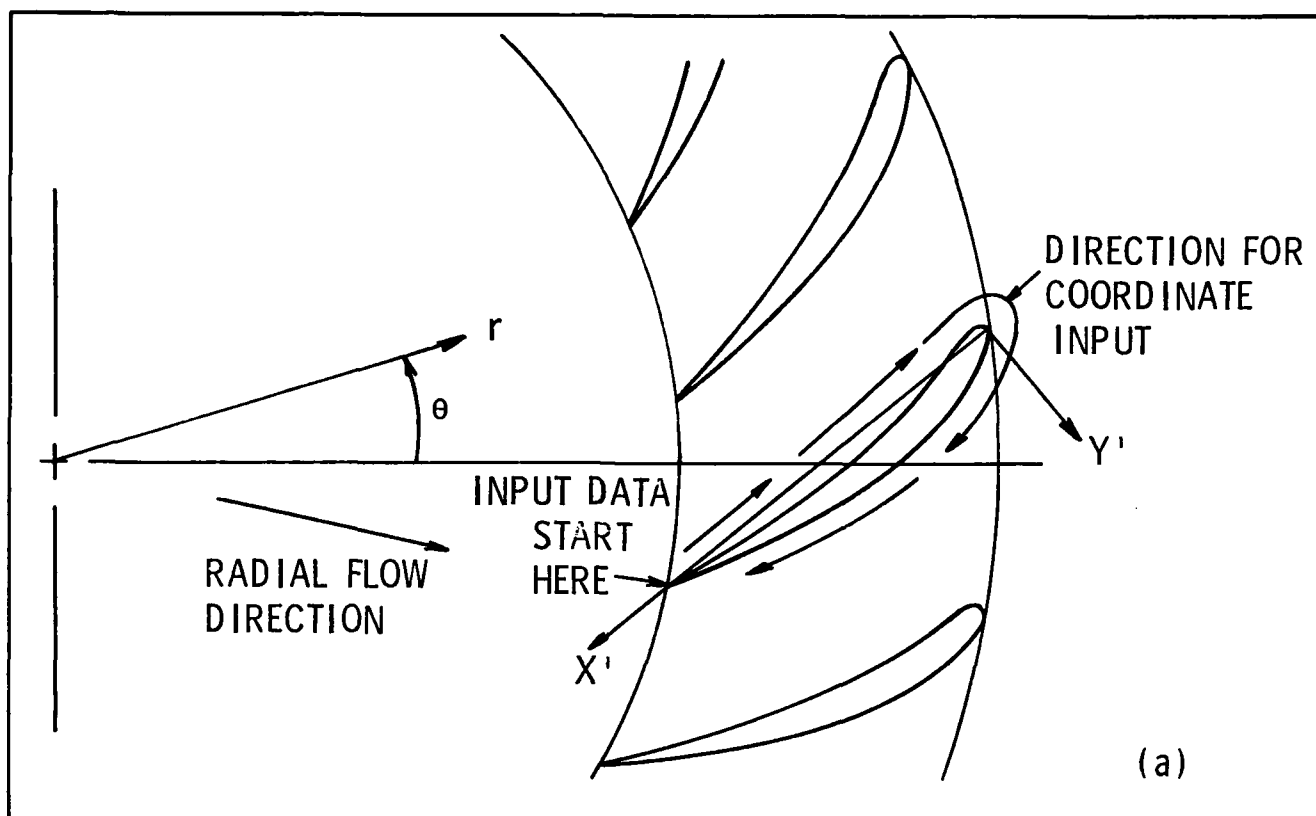


Figure C.4. a. Schematic of a Circular Cascade with the Radial Flow Direction Outward. The $x'-y'$ Orientation is the Same as in Figure C.3a.
b. Final Configuration of the Cascade After the Transformation

DISTRIBUTION LIST FOR ARL UNCLASSIFIED TM 81-130 by A. M. Yocum, dated
26 June 1981.

Commander
Naval Sea Systems Command
Department of the Navy
Washington, DC 20362
Attn: Library

Code NSEA-09G32
(Copy Nos. 1 and 2)

Naval Sea Systems Command
Attn: S. M. Blazek
Code NSEA-05HB
(Copy No. 3)

Commanding Officer
Naval Underwater Systems Center
Newport, RI 02840
Attn: Library
Code 54
(Copy No. 4)

Commanding Officer
Naval Ocean Systems Center
San Diego, CA 92152
Attn: D. Nelson
Code 6342
(Copy No. 5)

Commander
David W. Taylor Naval Ship R&D Center
Department of the Navy
Bethesda, MD 20084
Attn: T. E. Brockett
Code 1544
(Copy No. 6)

David W. Taylor Naval Ship R&D Center
Attn: J. H. McCarthy
Code 154
(Copy No. 7)

Officer-in-Charge
David W. Taylor Naval Ship R&D Center
Department of the Navy
Annapolis Laboratory
Annapolis, MD 21402
Attn: E. R. Quandt
Code 272
(Copy No. 8)

David W. Taylor Naval Ship R&D Center
Attn: J. G. Stricker
Code 2721
(Copy No. 9)

David W. Taylor Naval Ship R&D Center
Attn: J. Henry, IV
Code 2741
(Copy No. 10)

Defense Technical Information Center
5010 Duke Street
Cameron Station
Alexandria, VA 22314
(Copy Nos. 11 through 16)

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: W. M. McNally
MS 5-9
(Copy No. 17)

Applied Physics Laboratory/
University of Washington (APL/UW)
1013 NE 40th Street
Seattle, WA 98105
(Copy No. 18)

Applied Research Laboratories/
University of Texas (ARL/UT)
Austin, Texas 78712
(Copy No. 19)

Dr. Bruce D. Cox
Hydrodynamics Research Assoc., Inc.
7900 Inverness Ridge Road
Potomac, MD 20954
(Copy No. 20)

Allis-Chalmers Corp.
Hydro-Turbine Division
Box 712
Ycrk, PA 17405
Attn: R. K. Fisher
(Copy Nos. 21 and 22)

Dr. Henry T. Falvey
Engineering & Research Center
Bureau of Reclamation
U.S. Department of the Interior
Room 28, Bldg. 56
P.O. Box 25007
Denver Federal Center
Denver, CO 80225
(Copy No. 23)

DISTRIBUTION LIST FOR ARL UNCLASSIFIED TM 81-130 by A. M. Yocum, dated
26 June 1981.

Dr. Allen Moore
Admiralty Research Laboratory
Teddington, Middlesex
England
(Copy No. 24)

J. Lewis
University of Newcastle
Newcastle
England
(Copy No. 25)

Dr. John H. Horlock
Vice-Chancellor
The Open University
Walton Hall
MILTON KEYNES
MK7 6AA
England
(Copy No. 26)

Von Karman Inst. for Fluid Dynamics
Turbomachinery Laboratory
Rhode-Saint-Genese
Belgium
Attn: Library
(Copy No. 27)

Whittle Turbomachinery Laboratory
Madingley Road
Cambridge, England
Attn: Dr. D. S. Whitehead
(Copy No. 28)

David Wilson
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
(Copy No. 29)

Massachusetts Institute of Technology
Attn: Justin Kerwin
(Copy No. 30)

Dr. T. Oksihi
Mechanical Engineering Department
Iowa State University
Ames, Iowa 50010
(Copy No. 31)

Naval Postgraduate School
The Presidio
Monterey, CA 93940
Attn: Library
(Copy No. 32)

Creare, Inc.
Box 71
Hanover, NH 03755
Attn: W. Swift
(Copy No. 33)

The Pennsylvania State University
Applied Research Laboratory
Post Office Box 30
State College, PA 16801
Attn: W. S. Gearhart
(Copy No. 34)

Applied Research Laboratory
Attn: R. E. Henderson
(Copy No. 35)

B. Lakshminarayana
153H Hammond Building
University Park, PA 16802
(Copy No. 35)

